

HP StorageWorks 2/8q Fibre Channel Switch Simple Network Management Protocol

First Edition (November 2004)

Part Number: A7450-96008

This guide describes the support for Simple Network Management Protocol (SNMP) used with the HP StorageWorks 2/8q Fibre Channel (FC) Switch. This guide describes how to use SNMP to manage and monitor the switch.

© Copyright 2004 Hewlett-Packard Development Company, L.P.

Hewlett-Packard Company makes no warranty of any kind with regard to this material, including, but not limited to, the implied warranties of merchantability and fitness for a particular purpose. Hewlett-Packard shall not be liable for errors contained herein or for incidental or consequential damages in connection with the furnishing, performance, or use of this material.

This document contains proprietary information, which is protected by copyright. No part of this document may be photocopied, reproduced, or translated into another language without the prior written consent of Hewlett-Packard. The information contained in this document is subject to change without notice. The only warranties for HP products and services are set forth in the express warranty statements accompanying such products and services. Nothing herein should be construed as constituting an additional warranty. HP shall not be liable for technical or editorial errors or omissions contained herein.

Hewlett-Packard Company shall not be liable for technical or editorial errors or omissions contained herein. The information is provided "as is" without warranty of any kind and is subject to change without notice. The warranties for Hewlett-Packard Company products are set forth in the express limited warranty statements for such products. Nothing herein should be construed as constituting an additional warranty.

HP StorageWorks 2/8q Fibre Channel Switch Simple Network Management Protocol Reference Guide
First Edition (November 2004)
Part Number: A7450-96008

Contents

About this Guide.	15
Overview.	16
Intended audience.	16
Related documentation.	16
Conventions	17
Document conventions.	17
Text symbols	17
Equipment symbols	18
Rack stability	19
Getting help	20
HP installation and configuration assistance	20
HP technical support	20
HP storage web site	21
HP authorized reseller	21
1 SNMP Overview	23
SNMP interface objectives.	24
Manager and Agent.	25
Traps	26
Management Information Base.	26
User Datagram Protocol	27
Numbering system conventions	27
2 Configuring a Switch.	29
System specifications and requirements.	30
Configuring a switch using the Command Line Interface	31
Configuring a switch using Switch Manager	33
3 MIB-II Objects	35
Groups in MIB-II	36
System group	37
sysDescr (1.3.6.1.2.1.1.1).	37
sysObjectID (1.3.6.1.2.1.1.2).	37
sysUpTime (1.3.6.1.2.1.1.3).	37
sysContact (1.3.6.1.2.1.1.4)	38
sysName (1.3.6.1.2.1.1.5)	38
sysLocation (1.3.6.1.2.1.1.6)	38

sysServices (1.3.6.1.2.1.1.7)	38
Interfaces group	39
ifNumber (1.3.6.1.2.1.2.1)	39
Interfaces table	39
ifIndex (1.3.6.1.2.1.2.2.1.1)	39
ifDescr (1.3.6.1.2.1.2.2.1.2)	39
ifType (1.3.6.1.2.1.2.2.1.3)	39
ifMtu (1.3.6.1.2.1.2.2.1.4)	40
ifSpeed (1.3.6.1.2.1.2.2.1.5)	40
ifPhysAddress (1.3.6.1.2.1.2.2.1.6)	40
ifAdminStatus (1.3.6.1.2.1.2.2.1.7)	40
ifOperStatus (1.3.6.1.2.1.2.2.1.8)	40
ifLastChange (1.3.6.1.2.1.2.2.1.9)	41
ifInOctets (1.3.6.1.2.1.2.2.1.10)	41
ifInUcastPkts (1.3.6.1.2.1.2.2.1.11)	41
ifInNUcastPkts (1.3.6.1.2.1.2.2.1.12)	41
ifInDiscards (1.3.6.1.2.1.2.2.1.13)	41
ifInErrors (1.3.6.1.2.1.2.2.1.14)	41
ifInUnknownProtos (1.3.6.1.2.1.2.2.1.15)	42
ifOutOctets (1.3.6.1.2.1.2.2.1.16)	42
ifOutUcastPkts (1.3.6.1.2.1.2.2.1.17)	42
ifOutNUcastPkts (1.3.6.1.2.1.2.2.1.18)	42
ifOutDiscards (1.3.6.1.2.1.2.2.1.19)	42
ifOutErrors (1.3.6.1.2.1.2.2.1.20)	42
ifOutQLen (1.3.6.1.2.1.2.2.1.21)	43
ifSpecific (1.3.6.1.2.1.2.2.1.22)	43
Address translation group	44
atIfIndex (1.3.6.1.2.1.3.1.1.1)	44
atPhysAddress (1.3.6.1.2.1.3.1.1.2)	44
atNetAddress (1.3.6.1.2.1.3.1.1.3)	44
IP group	45
ipForwarding (1.3.6.1.2.1.4.1)	45
ipDefaultTTL (1.3.6.1.2.1.4.2)	45
ipInReceives (1.3.6.1.2.1.4.3)	45
ipInHdrErrors (1.3.6.1.2.1.4.4)	45
ipInAddrErrors (1.3.6.1.2.1.4.5)	46
ipForwDatagrams (1.3.6.1.2.1.4.6)	46
ipInUnknownProtos (1.3.6.1.2.1.4.7)	46
ipInDiscards (1.3.6.1.2.1.4.8)	46
ipInDelivers (1.3.6.1.2.1.4.9)	46
ipOutRequests (1.3.6.1.2.1.4.10)	47
ipOutDiscards (1.3.6.1.2.1.4.11)	47
ipOutNoRoutes (1.3.6.1.2.1.4.12)	47
ipReasmTimeout (1.3.6.1.2.1.4.13)	47
ipReasmReqds (1.3.6.1.2.1.4.14)	47
ipReasmOKs (1.3.6.1.2.1.4.15)	48
ipReasmFails (1.3.6.1.2.1.4.16)	48
ipFragOKs (1.3.6.1.2.1.4.17)	48
ipFragFails (1.3.6.1.2.1.4.18)	48
ipFragCreates (1.3.6.1.2.1.4.19)	48

IP address table	49
ipAdEntAddr (1.3.6.1.2.1.4.20.1.1)	49
ipAdEntIfIndex (1.3.6.1.2.1.4.20.1.2)	49
ipAdEntNetMask (1.3.6.1.2.1.4.20.1.3)	49
ipAdEntBcastAddr (1.3.6.1.2.1.4.20.1.4)	49
ipAdEntReasmMaxSize (1.3.6.1.2.1.4.20.1.5)	49
IP routing table	50
ipRouteDest (1.3.6.1.2.1.4.21.1.1)	50
ipRouteIfIndex (1.3.6.1.2.1.4.21.1.2)	50
ipRouteMetric1 (1.3.6.1.2.1.4.21.1.3)	50
ipRouteMetric2 (1.3.6.1.2.1.4.21.1.4)	50
ipRouteMetric3 (1.3.6.1.2.1.4.21.1.5)	51
ipRouteMetric4 (1.3.6.1.2.1.4.21.1.6)	51
ipRouteNextHop (1.3.6.1.2.1.4.21.1.7)	51
ipRouteType (1.3.6.1.2.1.4.21.1.8)	51
ipRouteProto (1.3.6.1.2.1.4.21.1.9)	52
ipRouteAge (1.3.6.1.2.1.4.21.1.10)	52
ipRouteMask (1.3.6.1.2.1.4.21.1.11)	52
ipRouteMetric5 (1.3.6.1.2.1.4.21.1.12)	52
ipRouteInfo (1.3.6.1.2.1.4.21.1.13)	52
IP address translation table	53
ipNetToMediaIfIndex (1.3.6.1.2.1.4.22.1.1)	53
ipNetToMediaPhysAddress (1.3.6.1.2.1.4.22.1.2)	53
ipNetToMediaNetAddress (1.3.6.1.2.1.4.22.1.3)	53
ipNetToMediaType (1.3.6.1.2.1.4.22.1.4)	53
Additional IP objects	54
ipRoutingDiscards (1.3.6.1.2.1.4.23)	54
ICMP group	55
icmpInMsgs (1.3.6.1.2.1.5.1)	55
icmpInErrors (1.3.6.1.2.1.5.2)	55
icmpInDestUnreachs (1.3.6.1.2.1.5.3)	55
icmpInTimeExcds (1.3.6.1.2.1.5.4)	55
icmpInParmProbs (1.3.6.1.2.1.5.5)	55
icmpInSrcQuenchs (1.3.6.1.2.1.5.6)	56
icmpInRedirects (1.3.6.1.2.1.5.7)	56
icmpInEchos (1.3.6.1.2.1.5.8)	56
icmpInEchoReps (1.3.6.1.2.1.5.9)	56
icmpInTimestamps (1.3.6.1.2.1.5.10)	56
icmpInTimestampReps (1.3.6.1.2.1.5.11)	56
icmpInAddrMasks (1.3.6.1.2.1.5.12)	57
icmpInAddrMaskReps (1.3.6.1.2.1.5.13)	57
icmpOutMsgs (1.3.6.1.2.1.5.14)	57
icmpOutErrors (1.3.6.1.2.1.5.15)	57
icmpOutDestUnreachs (1.3.6.1.2.1.5.16)	57
icmpOutTimeExcds (1.3.6.1.2.1.5.17)	57
icmpOutParmProbs (1.3.6.1.2.1.5.18)	58
icmpOutSrcQuenchs (1.3.6.1.2.1.5.19)	58
icmpOutRedirects (1.3.6.1.2.1.5.20)	58
icmpOutEchos (1.3.6.1.2.1.5.21)	58
icmpOutEchoReps (1.3.6.1.2.1.5.22)	58

icmpOutTimestamps (1.3.6.1.2.1.5.23)	58
icmpOutTimestampReps (1.3.6.1.2.1.5.24)	59
icmpOutAddrMasks (1.3.6.1.2.1.5.25)	59
icmpOutAddrMaskReps (1.3.6.1.2.1.5.26)	59
TCP group	60
tcpRtoAlgorithm (1.3.6.1.2.1.6.1)	60
tcpRtoMin (1.3.6.1.2.1.6.2)	60
tcpRtoMax (1.3.6.1.2.1.6.3)	60
tcpMaxConn (1.3.6.1.2.1.6.4)	60
tcpActiveOpens (1.3.6.1.2.1.6.5)	61
tcpPassiveOpens (1.3.6.1.2.1.6.6)	61
tcpAttemptFails (1.3.6.1.2.1.6.7)	61
tcpEstabResets (1.3.6.1.2.1.6.8)	61
tcpCurrEstab (1.3.6.1.2.1.6.9)	61
tcpInSegs (1.3.6.1.2.1.6.10)	62
tcpOutSegs (1.3.6.1.2.1.6.11)	62
tcpRetransSegs (1.3.6.1.2.1.6.12)	62
TCP connection table	62
tcpConnState (1.3.6.1.2.1.6.13.1.1)	62
tcpConnLocalAddress (1.3.6.1.2.1.6.13.1.2)	63
tcpConnLocalPort (1.3.6.1.2.1.6.13.1.3)	63
tcpConnRemAddress (1.3.6.1.2.1.6.13.1.4)	63
tcpConnRemPort (1.3.6.1.2.1.6.13.1.5)	63
Additional TCP objects	63
tcpInErrs (1.3.6.1.2.1.6.14)	63
tcpOutRsts (1.3.6.1.2.1.6.15)	63
UDP group	64
udpInDatagrams (1.3.6.1.2.1.7.1)	64
udpNoPorts (1.3.6.1.2.1.7.2)	64
udpInErrors (1.3.6.1.2.1.7.3)	64
udpOutDatagrams (1.3.6.1.2.1.7.4)	64
UDP listener table	64
udpLocalAddress (1.3.6.1.2.1.7.5.1.1)	64
udpLocalPort (1.3.6.1.2.1.7.5.1.2)	65
EGP group	66
egpInMsgs (1.3.6.1.2.1.8.1)	66
egpInErrors (1.3.6.1.2.1.8.2)	66
egpOutMsgs (1.3.6.1.2.1.8.3)	66
egpOutErrors (1.3.6.1.2.1.8.4)	66
EGP neighbor table	66
egpNeighState (1.3.6.1.2.1.8.5.1.1)	66
egpNeighAddr (1.3.6.1.2.1.8.5.1.2)	67
egpNeighAs (1.3.6.1.2.1.8.5.1.3)	67
egpNeighInMsgs (1.3.6.1.2.1.8.5.1.4)	67
egpNeighInErrs (1.3.6.1.2.1.8.5.1.5)	67
egpNeighOutMsgs (1.3.6.1.2.1.8.5.1.6)	67
egpNeighOutErrs (1.3.6.1.2.1.8.5.1.7)	67
egpNeighInErrMsgs (1.3.6.1.2.1.8.5.1.8)	68
egpNeighOutErrMsgs (1.3.6.1.2.1.8.5.1.9)	68
egpNeighStateUps (1.3.6.1.2.1.8.5.1.10)	68

egpNeighStateDowns (1.3.6.1.2.1.8.5.1.11)	68
egpNeighIntervalHello (1.3.6.1.2.1.8.5.1.12)	68
egpNeighIntervalPoll (1.3.6.1.2.1.8.5.1.13)	68
egpNeighMode (1.3.6.1.2.1.8.5.1.14)	69
egpNeighEventTrigger (1.3.6.1.2.1.8.5.1.15)	69
egpAs (1.3.6.1.2.1.8.6)	69
Transmission group	70
SNMP group	71
snmpInPkts (1.3.6.1.2.1.11.1)	71
snmpOutPkts (1.3.6.1.2.1.11.2)	71
snmpInBadVersions (1.3.6.1.2.1.11.3)	71
snmpInBadCommunityNames (1.3.6.1.2.1.11.4)	71
snmpInBadCommunityUses (1.3.6.1.2.1.11.5)	72
snmpInASNParseErrs (1.3.6.1.2.1.11.6)	72
snmpInTooBigs (1.3.6.1.2.1.11.8)	72
snmpInNoSuchNames (1.3.6.1.2.1.11.9)	72
snmpInBadValues (1.3.6.1.2.1.11.10)	72
snmpInReadOnlyls (1.3.6.1.2.1.11.11)	73
snmpInGenErrs (1.3.6.1.2.1.11.12)	73
snmpInTotalReqVars (1.3.6.1.2.1.11.13)	73
snmpInTotalSetVars (1.3.6.1.2.1.11.14)	73
snmpInGetRequests (1.3.6.1.2.1.11.15)	73
snmpInGetNexts (1.3.6.1.2.1.11.16)	74
snmpInSetRequests (1.3.6.1.2.1.11.17)	74
snmpInGetResponses (1.3.6.1.2.1.11.18)	74
snmpInTraps (1.3.6.1.2.1.11.19)	74
snmpOutTooBigs (1.3.6.1.2.1.11.20)	74
snmpOutNoSuchNames (1.3.6.1.2.1.11.21)	75
snmpOutBadValues (1.3.6.1.2.1.11.22)	75
snmpOutGenErrs (1.3.6.1.2.1.11.24)	75
snmpOutGetRequests (1.3.6.1.2.1.11.25)	75
snmpOutGetNexts (1.3.6.1.2.1.11.26)	75
snmpOutSetRequests (1.3.6.1.2.1.11.27)	75
snmpOutGetResponses (1.3.6.1.2.1.11.28)	76
snmpOutTraps (1.3.6.1.2.1.11.29)	76
snmpEnableAuthenTraps (1.3.6.1.2.1.11.30)	76
4 Fibre Alliance MIB Objects.	77
FA MIB definitions.	78
revisionNumber	79
Connectivity unit group	80
uNumber (1.3.6.1.3.94.1.1)	80
systemURL (1.3.6.1.3.94.1.2)	80
statusChangeTime (1.3.6.1.3.94.1.3)	80
configurationChangeTime (1.3.6.1.3.94.1.4)	80
connUnitTableChangeTime (1.3.6.1.3.94.1.5)	81
Connectivity table.	82
connUnitId (1.3.6.1.3.94.1.6.1.1)	82
connUnitGlobalId (1.3.6.1.3.94.1.6.1.2)	82
connUnitType (1.3.6.1.3.94.1.6.1.3)	82

connUnitNumports (1.3.6.1.3.94.1.6.1.4)	82
connUnitState (1.3.6.1.3.94.1.6.1.5)	83
connUnitStatus (1.3.6.1.3.94.1.6.1.6)	83
connUnitProduct (1.3.6.1.3.94.1.6.1.7)	84
connUnitSn (1.3.6.1.3.94.1.6.1.8)	84
connUnitUpTime (1.3.6.1.3.94.1.6.1.9)	84
connUnitUrl (1.3.6.1.3.94.1.6.1.10)	84
connUnitDomainId (1.3.6.1.3.94.1.6.1.11)	85
connUnitProxyMaster (1.3.6.1.3.94.1.6.1.12)	85
connUnitPrincipal (1.3.6.1.3.94.1.6.1.13)	85
connUnitNumSensors (1.3.6.1.3.94.1.6.1.14)	85
connUnitStatusChangeTime (1.3.6.1.3.94.1.6.1.15)	86
connUnitConfigurationChangeTime (1.3.6.1.3.94.1.6.1.16)	86
connUnitNumRevs (1.3.6.1.3.94.1.6.1.17)	86
connUnitNumZones (1.3.6.1.3.94.1.6.1.18)	86
connUnitModuleId (1.3.6.1.3.94.1.6.1.19)	87
connUnitName (1.3.6.1.3.94.1.6.1.20)	87
connUnitInfo (1.3.6.1.3.94.1.6.1.21)	87
connUnitControl (1.3.6.1.3.94.1.6.1.22)	87
connUnitContact (1.3.6.1.3.94.1.6.1.23)	88
connUnitLocation (1.3.6.1.3.94.1.6.1.24)	88
connUnitEventFilter (1.3.6.1.3.94.1.6.1.25)	89
connUnitNumEvents (1.3.6.1.3.94.1.6.1.26)	89
connUnitMaxEvents (1.3.6.1.3.94.1.6.1.27)	90
connUnitEventCurrID (1.3.6.1.3.94.1.6.1.28)	90
connUnitFabricID (1.3.6.1.3.94.1.6.1.29)	90
connUnitNumLinks (1.3.6.1.3.94.1.6.1.30)	90
connUnitVendorId (1.3.6.1.3.94.1.6.1.31)	90
Revision table	91
connUnitRevsUnitId (1.3.6.1.3.94.1.7.1.1)	91
connUnitRevsIndex (1.3.6.1.3.94.1.7.1.2)	91
connUnitRevsRevId (1.3.6.1.3.94.1.7.1.3)	91
connUnitRevsDescription (1.3.6.1.3.94.1.7.1.4)	92
Sensor table	93
connUnitSensorUnitId (1.3.6.1.3.94.1.8.1.1)	93
connUnitSensorIndex (1.3.6.1.3.94.1.8.1.2)	93
connUnitSensorName (1.3.6.1.3.94.1.8.1.3)	93
connUnitSensorStatus (1.3.6.1.3.94.1.8.1.4)	94
connUnitSensorInfo (1.3.6.1.3.94.1.8.1.5)	94
connUnitSensorMessage (1.3.6.1.3.94.1.8.1.6)	95
connUnitSensorType (1.3.6.1.3.94.1.8.1.7)	96
connUnitSensorCharacteristic (1.3.6.1.3.94.1.8.1.8)	96
Port table	97
connUnitPortUnitId (1.3.6.1.3.94.1.10.1.1)	97
connUnitPortIndex (1.3.6.1.3.94.1.10.1.2)	97
connUnitPortType (1.3.6.1.3.94.1.10.1.3)	97
connUnitPortFCClassCap (1.3.6.1.3.94.1.10.1.4)	98
connUnitPortFCClassOp (1.3.6.1.3.94.1.10.1.5)	98
connUnitPortState (1.3.6.1.3.94.1.10.1.5)	99
connUnitPortStatus (1.3.6.1.3.94.1.10.1.7)	99

connUnitPortTransmitterType (1.3.6.1.3.94.1.10.1.8)	99
connUnitPortModuleType (1.3.6.1.3.94.1.10.1.9)	100
connUnitPortWwn (1.3.6.1.3.94.1.10.1.10)	100
connUnitPortFCId (1.3.6.1.3.94.1.10.1.11)	101
connUnitPortSn (1.3.6.1.3.94.1.10.1.12)	101
connUnitPortRevision (1.3.6.1.3.94.1.10.1.13)	101
connUnitPortVendor (1.3.6.1.3.94.1.10.1.14)	101
connUnitPortSpeed (1.3.6.1.3.94.1.10.1.15)	101
connUnitPortControl (1.3.6.1.3.94.1.10.1.16)	102
connUnitPortName (1.3.6.1.3.94.1.10.1.17)	103
connUnitPortPhysicalNumber (1.3.6.1.3.94.1.10.1.18)	103
connUnitPortStatObject (1.3.6.1.3.94.1.10.1.19)	103
connUnitPortProtocolCap (1.3.6.1.3.94.1.10.1.20)	104
connUnitPortProtocolOp (1.3.6.1.3.94.1.10.1.21)	104
connUnitPortNodeWwn (1.3.6.1.3.94.1.10.1.22)	104
connUnitPortHWState (1.3.6.1.3.94.1.10.1.23)	105
Event table	106
connUnitEventUnitId (1.3.6.1.3.94.1.11.1.1)	106
connUnitEventIndex (1.3.6.1.3.94.1.11.1.2)	106
connUnitEventId (1.3.6.1.3.94.1.11.1.3)	107
connUnitREventTime (1.3.6.1.3.94.1.11.1.4)	107
connUnitSEventTime (1.3.6.1.3.94.1.11.1.5)	107
connUnitEventSeverity (1.3.6.1.3.94.1.11.1.6)	108
connUnitEventType (1.3.6.1.3.94.1.11.1.7)	109
connUnitEventObject (1.3.6.1.3.94.1.11.1.8)	109
connUnitEventDescr (1.3.6.1.3.94.1.11.1.9)	109
Link table	110
connUnitLinkUnitId (1.3.6.1.3.94.1.12.1.1)	110
connUnitLinkIndex (1.3.6.1.3.94.1.12.1.2)	110
connUnitLinkNodeIdX (1.3.6.1.3.94.1.12.1.3)	111
connUnitLinkPortNumberX (1.3.6.1.3.94.1.12.1.4)	111
connUnitLinkPortWwnX (1.3.6.1.3.94.1.12.1.5)	111
connUnitLinkNodeIdY (1.3.6.1.3.94.1.12.1.6)	111
connUnitLinkPortNumberY (1.3.6.1.3.94.1.12.1.7)	112
connUnitLinkPortWwnY (1.3.6.1.3.94.1.12.1.8)	112
connUnitLinkAgentAddressY (1.3.6.1.3.94.1.12.1.9)	112
connUnitLinkAgentAddressTypeY (1.3.6.1.3.94.1.12.1.10)	112
connUnitLinkAgentPortY (1.3.6.1.3.94.1.12.1.11)	113
connUnitLinkUnitTypeY (1.3.6.1.3.94.1.12.1.12)	113
connUnitLinkConnIdY (1.3.6.1.3.94.1.12.1.13)	113
connUnitLinkCurrIndex (1.3.6.1.3.94.1.12.1.14)	113
Zone table	114
connUnitZoneIndex (1.3.6.1.3.94.1.13.1.1)	114
connUnitZoneMemberIndex (1.3.6.1.3.94.1.13.1.2)	114
connUnitZoneSetName (1.3.6.1.3.94.1.13.1.3)	114
connUnitZoneSetNumZones (1.3.6.1.3.94.1.13.1.4)	114
connUnitZoneName (1.3.6.1.3.94.1.13.1.5)	115
connUnitZoneCapabilities (1.3.6.1.3.94.1.13.1.6)	115
connUnitZoneEnforcementState (1.3.6.1.3.94.1.13.1.7)	115
connUnitZoneAttributeBlock (1.3.6.1.3.94.1.13.1.8)	115

connUnitZoneNumMembers (1.3.6.1.3.94.1.13.1.9)	116
connUnitZoneMemberIdType (1.3.6.1.3.94.1.13.1.10)	116
connUnitZoneMemberID (1.3.6.1.3.94.1.13.1.11)	116
Zoning alias table	117
connUnitZoningAliasIndex (1.3.6.1.3.94.1.14.1.1)	117
connUnitZoningAliasMemberIndex (1.3.6.1.3.94.1.14.1.2)	117
connUnitZoningAliasNumAliases (1.3.6.1.3.94.1.14.1.3)	117
connUnitZoningAliasName (1.3.6.1.3.94.1.14.1.4)	117
connUnitZoningAliasNumMembers (1.3.6.1.3.94.1.14.1.5)	118
connUnitZoningAliasMemberIdType (1.3.6.1.3.94.1.14.1.6)	118
connUnitZoningAliasMemberID (1.3.6.1.3.94.1.14.1.7)	118
Port statistics table	119
connUnitPortStatUnitId (1.3.6.1.3.94.4.5.1.1)	119
connUnitPortStatIndex (1.3.6.1.3.94.4.5.1.2)	119
connUnitPortStatCountError (1.3.6.1.3.94.4.5.1.3)	119
connUnitPortStatCountTxObjects (1.3.6.1.3.94.4.5.1.4)	120
connUnitPortStatCountRxObjects (1.3.6.1.3.94.4.5.1.5)	120
connUnitPortStatCountTxElements (1.3.6.1.3.94.4.5.1.6)	120
connUnitPortStatCountRxElements (1.3.6.1.3.94.4.5.1.7)	120
connUnitPortStatCountBBCreditZero (1.3.6.1.3.94.4.5.1.8)	121
connUnitPortStatCountInputBuffersFull (1.3.6.1.3.94.4.5.1.9)	121
connUnitPortStatCountFBSYFrames (1.3.6.1.3.94.4.5.1.10)	121
connUnitPortStatCountPBSYFrames (1.3.6.1.3.94.4.5.1.11)	121
connUnitPortStatCountFRJTFrames (1.3.6.1.3.94.4.5.1.12)	122
connUnitPortStatCountPRJTFrames (1.3.6.1.3.94.4.5.1.13)	122
connUnitPortStatCountClass1RxFrames (1.3.6.1.3.94.4.5.1.14)	122
connUnitPortStatCountClass1TxFrames (1.3.6.1.3.94.4.5.1.15)	122
connUnitPortStatCountClass1FBSYFrames (1.3.6.1.3.94.4.5.1.16)	123
connUnitPortStatCountClass1PBSYFrames (1.3.6.1.3.94.4.5.1.17)	123
connUnitPortStatCountClass1FRJTFrames (1.3.6.1.3.94.4.5.1.18)	123
connUnitPortStatCountClass1PRJTFrames (1.3.6.1.3.94.4.5.1.19)	123
connUnitPortStatCountClass2RxFrames (1.3.6.1.3.94.4.5.1.20)	124
connUnitPortStatCountClass2TxFrames (1.3.6.1.3.94.4.5.1.21)	124
connUnitPortStatCountClass2FBSYFrames (1.3.6.1.3.94.4.5.1.22)	124
connUnitPortStatCountClass2PBSYFrames (1.3.6.1.3.94.4.5.1.23)	124
connUnitPortStatCountClass2FRJTFrames (1.3.6.1.3.94.4.5.1.24)	125
connUnitPortStatCountClass2PRJTFrames (1.3.6.1.3.94.4.5.1.25)	125
connUnitPortStatCountClass3RxFrames (1.3.6.1.3.94.4.5.1.26)	125
connUnitPortStatCountClass3TxFrames (1.3.6.1.3.94.4.5.1.27)	125
connUnitPortStatCountClass3Discards (1.3.6.1.3.94.4.5.1.28)	125
connUnitPortStatCountRxMulticastObjects (1.3.6.1.3.94.4.5.1.29)	126
connUnitPortStatCountTxMulticastObjects (1.3.6.1.3.94.4.5.1.30)	126
connUnitPortStatCountRxBroadcastObjects (1.3.6.1.3.94.4.5.1.31)	126
connUnitPortStatCountTxBroadcastObjects (1.3.6.1.3.94.4.5.1.32)	126
connUnitPortStatCountRxLinkResets (1.3.6.1.3.94.4.5.1.33)	126
connUnitPortStatCountTxLinkResets (1.3.6.1.3.94.4.5.1.34)	127
connUnitPortStatCountNumberLinkResets (1.3.6.1.3.94.4.5.1.35)	127
connUnitPortStatCountRxOfflineSequences (1.3.6.1.3.94.4.5.1.36)	127
connUnitPortStatCountTxOfflineSequences (1.3.6.1.3.94.4.5.1.37)	127
connUnitPortStatCountNumberOfflineSequences (1.3.6.1.3.94.4.5.1.38)	127

connUnitPortStatCountLinkFailures (1.3.6.1.3.94.4.5.1.39)	128
connUnitPortStatCountInvalidCRC (1.3.6.1.3.94.4.5.1.40)	128
connUnitPortStatCountInvalidTxWords (1.3.6.1.3.94.4.5.1.41)	128
connUnitPortStatCountPrimitiveSequenceProtocolErrors (1.3.6.1.3.94.4.5.1.42)	128
connUnitPortStatCountLossofSignal (1.3.6.1.3.94.4.5.1.43)	129
connUnitPortStatCountLossofSynchronization (1.3.6.1.3.94.4.5.1.44)	129
connUnitPortStatCountInvalidOrderedSets (1.3.6.1.3.94.4.5.1.45)	129
connUnitPortStatCountFramesTooLong (1.3.6.1.3.94.4.5.1.46)	129
connUnitPortStatCountFramesTruncated (1.3.6.1.3.94.4.5.1.47)	130
connUnitPortStatCountAddressErrors (1.3.6.1.3.94.4.5.1.48)	130
connUnitPortStatCountDelimiterErrors (1.3.6.1.3.94.4.5.1.49)	130
connUnitPortStatCountEncodingDisparityErrors (1.3.6.1.3.94.4.5.1.50)	130
Simple Name Server (SNS) table	131
connUnitSnsMaxEntry (1.3.6.1.3.94.5.1.1)	131
connUnitSnsId (1.3.6.1.3.94.5.2.1.1.1)	131
connUnitSnsPortIndex (1.3.6.1.3.94.5.2.1.1.2)	131
connUnitSnsPortIdentifier (1.3.6.1.3.94.5.2.1.1.3)	132
connUnitSnsPortName (1.3.6.1.3.94.5.2.1.1.4)	132
connUnitSnsNodeName (1.3.6.1.3.94.5.2.1.1.5)	132
connUnitSnsClassOfSvc (1.3.6.1.3.94.5.2.1.1.6)	132
connUnitSnsNodeIPAddress (1.3.6.1.3.94.5.2.1.1.7)	132
connUnitSnsProcAssoc (1.3.6.1.3.94.5.2.1.1.8)	133
connUnitSnsFC4Type (1.3.6.1.3.94.5.2.1.1.9)	133
connUnitSnsPortType (1.3.6.1.3.94.5.2.1.1.10)	133
connUnitSnsPortIPAddress (1.3.6.1.3.94.5.2.1.1.11)	134
connUnitSnsFabricPortName (1.3.6.1.3.94.5.2.1.1.12)	134
connUnitSnsHardAddress (1.3.6.1.3.94.5.2.1.1.13)	134
connUnitSnsSymbolicPortName (1.3.6.1.3.94.5.2.1.1.14)	134
connUnitSnsSymbolicNodeName (1.3.6.1.3.94.5.2.1.1.15)	134
Platform table	135
connUnitPlatformMaxEntry (1.3.6.1.3.94.5.1.2)	135
connUnitPlatformIndex (1.3.6.1.3.94.5.2.2.1.1)	135
connUnitPlatformNodeIndex (1.3.6.1.3.94.5.2.2.1.2)	135
connUnitPlatformUnitID (1.3.6.1.3.94.5.2.2.1.3)	135
connUnitPlatformName (1.3.6.1.3.94.5.2.2.1.4)	136
connUnitPlatformType (1.3.6.1.3.94.5.2.2.1.6)	136
connUnitPlatformLabel (1.3.6.1.3.94.5.2.2.1.7)	136
connUnitPlatformDescription (1.3.6.1.3.94.5.2.2.1.8)	136
connUnitPlatformLocation (1.3.6.1.3.94.5.2.2.1.9)	136
connUnitPlatformManagementUrl (1.3.6.1.3.94.5.2.2.1.10)	137
connUnitPlatformNumNodes (1.3.6.1.3.94.5.2.2.1.11)	137
connUnitPlatformNodeName (1.3.6.1.3.94.5.2.2.1.12)	137
Trap table	138
trapMaxClients (1.3.6.1.3.94.2.1)	138
trapClientCount (1.3.6.1.3.94.2.2)	138
trapRegIpAddress (1.3.6.1.3.94.2.3.1.1)	139
trapRegPort (1.3.6.1.3.94.2.3.1.2)	139
trapRegFilter (1.3.6.1.3.94.2.3.1.3)	139
trapRegRowState (1.3.6.1.3.94.2.3.1.4)	139
Related traps	141

connUnitStatusChange (1.3.6.1.3.94.0.1)	141
connUnitDeletedTrap (1.3.6.1.3.94.0.3)	141
connUnitEventTrap (1.3.6.1.3.94.0.4)	141
connUnitSensorStatusChange (1.3.6.1.3.94.0.5)	143
connUnitPortStatusChange (1.3.6.1.3.94.0.6)	143
coldStart	143
authenticationFailure	144
5 Fabric Element MIB Objects	145
Fibre Channel FE MIB definitions	146
Configuration group	147
Scalar objects	147
fcFeFabricName (1.3.6.1.2.1.75.1.1.1)	147
fcFeElementName (1.3.6.1.2.1.75.1.1.2)	147
fcFeModuleCapacity (1.3.6.1.2.1.75.1.1.3)	147
Module table	148
fcFeModuleDescr (1.3.6.1.2.1.75.1.1.4.1.2)	148
fcFeModuleObjectID (1.3.6.1.2.1.75.1.1.4.1.3)	148
fcFeModuleOperStatus (1.3.6.1.2.1.75.1.1.4.1.4)	148
fcFeModuleLastChange (1.3.6.1.2.1.75.1.1.4.1.5)	149
fcFeModuleFxPortCapacity (1.3.6.1.2.1.75.1.1.4.1.6)	149
fcFeModuleName (1.3.6.1.2.1.75.1.1.4.1.7)	149
FxPort configuration table	150
fcFxPortName (1.3.6.1.2.1.75.1.1.5.1.2)	150
fcFxPortFcphVersionHigh (1.3.6.1.2.1.75.1.1.5.1.3)	150
fcFxPortFcphVersionLow (1.3.6.1.2.1.75.1.1.5.1.4)	150
fcFxPortBbCredit (1.3.6.1.2.1.75.1.1.5.1.5)	150
fcFxPortRxBufSize (1.3.6.1.2.1.75.1.1.5.1.6)	151
fcFxPortRatov (1.3.6.1.2.1.75.1.1.5.1.7)	151
fcFxPortEdtov (1.3.6.1.2.1.75.1.1.5.1.8)	151
fcFxPortCosSupported (1.3.6.1.2.1.75.1.1.5.1.9)	151
fcFxPortIntermixSupported (1.3.6.1.2.1.75.1.1.5.1.10)	151
fcFxPortStackedConnMode (1.3.6.1.2.1.75.1.1.5.1.11)	152
fcFxPortClass2SeqDeliv (1.3.6.1.2.1.75.1.1.5.1.12)	152
fcFxPortClass3SeqDeliv (1.3.6.1.2.1.75.1.1.5.1.13)	152
fcFxPortHoldTime (1.3.6.1.2.1.75.1.1.5.1.14)	152
Status group	153
FxPort status table	153
fcFxPortID (1.3.6.1.2.1.75.1.2.1.1.1)	153
fcFxPortBbCreditAvailable (1.3.6.1.2.1.75.1.2.1.1.2)	153
fcFxPortOperMode (1.3.6.1.2.1.75.1.2.1.1.3)	153
fcFxPortAdminMode (1.3.6.1.2.1.75.1.2.1.1.4)	154
FxPort physical level table	154
fcFxPortPhysAdminStatus (1.3.6.1.2.1.75.1.2.2.1.1)	154
fcFxPortPhysOperStatus (1.3.6.1.2.1.75.1.2.2.1.2)	155
fcFxPortPhysLastChange (1.3.6.1.2.1.75.1.2.2.1.3)	156
fcFxPortPhysRttov (1.3.6.1.2.1.75.1.2.2.1.4)	156
FxPort fabric login table	156
fcFxPortFcphVersionAgreed (1.3.6.1.2.1.75.1.2.3.1.2)	156
fcFxPortNxPortBbCredit (1.3.6.1.2.1.75.1.2.3.1.3)	156

fcFxFxPortNxPortRxDataFieldSize (1.3.6.1.2.1.75.1.2.3.1.4)	157
fcFxFxPortCosSuppAgreed (1.3.6.1.2.1.75.1.2.3.1.5)	157
fcFxFxPortIntermixSuppAgreed (1.3.6.1.2.1.75.1.2.3.1.6)	157
fcFxFxPortStackedConnModeAgreed (1.3.6.1.2.1.75.1.2.3.1.7)	157
fcFxFxPortClass2SeqDelivAgreed (1.3.6.1.2.1.75.1.2.3.1.8)	158
fcFxFxPortClass3SeqDelivAgreed (1.3.6.1.2.1.75.1.2.3.1.9)	158
fcFxFxPortNxPortName (1.3.6.1.2.1.75.1.2.3.1.10)	158
fcFxFxPortConnectedNxPort (1.3.6.1.2.1.75.1.2.3.1.11)	158
fcFxFxPortBbCreditModel (1.3.6.1.2.1.75.1.2.3.1.12)	158
Error group	159
FxFxPort Error Table	159
fcFxFxPortLinkFailures (1.3.6.1.2.1.75.1.3.1.1.1)	159
fcFxFxPortSyncLosses (1.3.6.1.2.1.75.1.3.1.1.2)	159
fcFxFxPortSigLosses (1.3.6.1.2.1.75.1.3.1.1.3)	159
fcFxFxPortPrimSeqProtoErrors (1.3.6.1.2.1.75.1.3.1.1.4)	159
fcFxFxPortInvalidTxWords (1.3.6.1.2.1.75.1.3.1.1.5)	160
fcFxFxPortInvalidCrcs (1.3.6.1.2.1.75.1.3.1.1.6)	160
fcFxFxPortDelimiterErrors (1.3.6.1.2.1.75.1.3.1.1.7)	160
fcFxFxPortAddressIdErrors (1.3.6.1.2.1.75.1.3.1.1.8)	160
fcFxFxPortLinkResetIns (1.3.6.1.2.1.75.1.3.1.1.9)	160
fcFxFxPortLinkResetOuts (1.3.6.1.2.1.75.1.3.1.1.10)	161
fcFxFxPortOlsIns (1.3.6.1.2.1.75.1.3.1.1.11)	161
fcFxFxPortOlsOuts (1.3.6.1.2.1.75.1.3.1.1.12)	161
Accounting groups	162
Class 1 accounting table	162
fcFxFxPortC1InFrames (1.3.6.1.2.1.75.1.4.1.1.1)	162
fcFxFxPortC1OutFrames (1.3.6.1.2.1.75.1.4.1.1.2)	162
fcFxFxPortC1InOctets (1.3.6.1.2.1.75.1.4.1.1.3)	162
fcFxFxPortC1OutOctets (1.3.6.1.2.1.75.1.4.1.1.4)	163
fcFxFxPortC1Discards (1.3.6.1.2.1.75.1.4.1.1.5)	163
fcFxFxPortC1FbsyFrames (1.3.6.1.2.1.75.1.4.1.1.6)	163
fcFxFxPortC1FrjtFrames (1.3.6.1.2.1.75.1.4.1.1.7)	163
fcFxFxPortC1InConnections (1.3.6.1.2.1.75.1.4.1.1.8)	163
fcFxFxPortC1OutConnections (1.3.6.1.2.1.75.1.4.1.1.9)	164
fcFxFxPortC1ConnTime (1.3.6.1.2.1.75.1.4.1.1.10)	164
Class 2 accounting table	164
fcFxFxPortC2InFrames (1.3.6.1.2.1.75.1.4.2.1.1)	164
fcFxFxPortC2OutFrames (1.3.6.1.2.1.75.1.4.2.1.2)	164
fcFxFxPortC2InOctets (1.3.6.1.2.1.75.1.4.2.1.3)	165
fcFxFxPortC2OutOctets (1.3.6.1.2.1.75.1.4.2.1.4)	165
fcFxFxPortC2Discards (1.3.6.1.2.1.75.1.4.2.1.5)	165
fcFxFxPortC2FbsyFrames (1.3.6.1.2.1.75.1.4.2.1.6)	165
fcFxFxPortC2FrjtFrames (1.3.6.1.2.1.75.1.4.2.1.7)	165
Class 3 accounting table	166
fcFxFxPortC3InFrames (1.3.6.1.2.1.75.1.4.3.1.1)	166
fcFxFxPortC3OutFrames (1.3.6.1.2.1.75.1.4.3.1.2)	166
fcFxFxPortC3InOctets (1.3.6.1.2.1.75.1.4.3.1.3)	166
fcFxFxPortC3OutOctets (1.3.6.1.2.1.75.1.4.3.1.4)	166
fcFxFxPortC3Discards (1.3.6.1.2.1.75.1.4.3.1.5)	167
Capability group	168

FxPort capability table	168
fcFxPortCapFcphVersionHigh (1.3.6.1.2.1.75.1.5.1.1.1)	168
fcFxPortCapFcphVersionLow (1.3.6.1.2.1.75.1.5.1.1.2)	168
fcFxPortCapBbCreditMax (1.3.6.1.2.1.75.1.5.1.1.3)	168
fcFxPortCapBbCreditMin (1.3.6.1.2.1.75.1.5.1.1.4)	168
fcFxPortCapRxDataFieldSizeMax (1.3.6.1.2.1.75.1.5.1.1.5)	169
fcFxPortCapRxDataFieldSizeMin (1.3.6.1.2.1.75.1.5.1.1.6)	169
fcFxPortCapCos (1.3.6.1.2.1.75.1.5.1.1.7)	169
fcFxPortCapIntermix (1.3.6.1.2.1.75.1.5.1.1.8)	169
fcFxPortCapStackedConnMode (1.3.6.1.2.1.75.1.5.1.1.9)	169
fcFxPortCapClass2SeqDeliv (1.3.6.1.2.1.75.1.5.1.1.10)	170
fcFxPortCapClass3SeqDeliv (1.3.6.1.2.1.75.1.5.1.1.11)	170
fcFxPortCapHoldTimeMaxv (1.3.6.1.2.1.75.1.5.1.1.12)	170
fcFxPortCapHoldTimeMin (1.3.6.1.2.1.75.1.5.1.1.13)	170
6 MIB Objects	171
MIB definitions	172
fcQxPortPhysAdminStatus (1.3.6.1.4.1.1663.1.3.10.1.1.3)	172
fcQxPortPhysOperStatus (1.3.6.1.4.1.1663.1.3.10.1.1.4)	173
Index	175

About This Guide

This guide describes the support for Simple Network Management Protocol (SNMP) used with HP StorageWorks 2/8q Fibre Channel Switch. This Simple Network Management Protocol (SNMP) Reference Guide describes how to use SNMP to manage and monitor the switch.

Chapter topics include:

- Chapter 1—Provides an overview of SNMP objectives, managers and agents, traps, Management Information Bases (MIB), and User Datagram Protocol.
- Chapter 2—Describes how to configure the switch using Telnet and the Switch Manager graphical user interface.
- Chapter 3—Describes the Management Information Database (MIB-II).
- Chapter 4—Describes the Fibre Alliance - Management Information Database (FA-MIB version 4.0).
- Chapter 5—Describes the Fabric Element - Management Information Database (FE-MIB).
- Chapter 6—Describes the implementation details.

Overview

This section covers the following topics:

- [Intended audience](#)
- [Related documentation](#)

Intended audience

This guide is intended for users responsible for the support of SNMP and the 2/8q FC Switch configurations.

Related documentation

The following MSA1000 SAN Kit and 2/8q FC Switch documents are on the MSA1000 Small Business SAN documentation CD:

- *HP StorageWorks Modular Smart Array 1000 Small Business SAN Kit Installation Instructions* (also printed)
- *HP StorageWorks 2/8q Fibre Channel Switch Management User Guide*
- *HP StorageWorks 2/8q Fibre Channel Switch Simple Network Management Protocol Reference Guide*
- *HP StorageWorks 2/8q Fibre Channel Switch Installation Guide*
- *HP StorageWorks 2/8q Fibre Channel Switch Event Messages Reference Guide*

The following MSA1000-specific documents are on the MSA1000 documentation CD:

- *HP StorageWorks MSA1000 Configuration Overview*
- *HP StorageWorks MSA1000 Installation Guide*
- *HP StorageWorks Modular Smart Array 1000 Maintenance and Service Guide*
- *HP StorageWorks Modular Smart Array 1000/1500 cs Command Line Interface User Guide*
- *HP StorageWorks Modular Smart Array 1000 Controller Reference Guide*
- *HP Array Configuration Utility User Guide*

Conventions

Conventions consist of the following:

- Document conventions
- Text symbols
- Equipment symbols

Document conventions

This document follows the conventions in [Table 1](#).

Table 1: Document conventions

Convention	Element
Blue text: Figure 1	Cross-reference links
Bold	Menu items, buttons, and key, tab, and box names
<i>Italics</i>	Text emphasis and document titles in body text
Monospace font	User input, commands, code, file and directory names, and system responses (output and messages)
<i>Monospace, italic font</i>	Command-line and code variables
Blue underlined sans serif font text (http://www.hp.com)	Web site addresses

Text symbols

The following symbols may be found in the text of this guide. They have the following meanings:



WARNING: Text set off in this manner indicates that failure to follow directions in the warning could result in bodily harm or death.



Caution: Text set off in this manner indicates that failure to follow directions could result in damage to equipment or data.

Tip: Text in a tip provides additional help to readers by providing nonessential or optional techniques, procedures, or shortcuts.

Note: Text set off in this manner presents commentary, sidelights, or interesting points of information.

Equipment symbols

The following equipment symbols may be found on hardware for which this guide pertains. They have the following meanings:



Any enclosed surface or area of the equipment marked with these symbols indicates the presence of electrical shock hazards. Enclosed area contains no operator serviceable parts.

WARNING: To reduce the risk of personal injury from electrical shock hazards, do not open this enclosure.



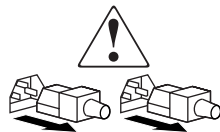
Any RJ-45 receptacle marked with these symbols indicates a network interface connection.

WARNING: To reduce the risk of electrical shock, fire, or damage to the equipment, do not plug telephone or telecommunications connectors into this receptacle.



Any surface or area of the equipment marked with these symbols indicates the presence of a hot surface or hot component. Contact with this surface could result in injury.

WARNING: To reduce the risk of personal injury from a hot component, allow the surface to cool before touching.



Power supplies or systems marked with these symbols indicate the presence of multiple sources of power.

WARNING: To reduce the risk of personal injury from electrical shock, remove all power cords to completely disconnect power from the power supplies and systems.



Any product or assembly marked with these symbols indicates that the component exceeds the recommended weight for one individual to handle safely.

WARNING: To reduce the risk of personal injury or damage to the equipment, observe local occupational health and safety requirements and guidelines for manually handling material.

Getting help

If you still have a question after reading this guide, contact an HP Authorized Service Provider or access our web site: <http://www.hp.com>.

Note: HP call centers use product and serial numbers to validate warranty entitlement. Most HP products can provide product number, serial number and firmware revision electronically through the use of supplied management or diagnostic utilities, eliminating the need to physically inspect or remove products from installed enclosures. You may be directed by HP to run these utilities to gather required entitlement information.

HP installation and configuration assistance

A moderate level of SAN-related knowledge is required to successfully install this product. If you are not familiar with installing and configuring storage array systems in a SAN, HP can install it for you. For more information, access our web site at http://www.hp.com/hps/storage/ns_implementation.html.

Depending on your needs, different levels of assistance are available. For example, the HP Installation and Startup for HP StorageWorks Disk Arrays Service Package includes:

- Physical installation of the MSA.
- Virtual disk design and configuration of the MSA.
- Service planning.
- Service deployment.
- Installation Verification Testing (IVT).
- Customer orientation.

HP technical support

Telephone numbers for worldwide technical support are listed on the following HP web site: <http://www.hp.com/support/>. From this web site, select the country of origin.

Note: For continuous quality improvement, calls may be recorded or monitored.

Be sure to have the following information available before calling:

- Technical support registration number (if applicable)
- Product serial numbers
- Product model names and numbers
- Applicable error messages
- Operating system type and revision level
- Detailed, specific questions

HP storage web site

The HP web site has the latest information on this product, as well as the latest drivers. Access storage at: <http://www.hp.com/country/us/eng/prodserv/storage.html>. From this web site, select the appropriate product or solution.

HP authorized reseller

For the name of your nearest HP Authorized Reseller:

- In the United States, call 1-800-345-1518
- In Canada, call 1-800-263-5868
- Elsewhere, see the HP web site for locations and telephone numbers: <http://www.hp.com>.

SNMP Overview

1

Simple Network Management Protocol (SNMP) is the protocol governing network management and monitoring of network devices. This SNMP Reference Guide describes how to use SNMP to manage and monitor the switch. Specifically, this guide describes the SNMP agent that resides on the switch.

The following topics are covered in this chapter:

- [SNMP interface objectives](#), page 20
- [Manager and Agent](#), page 21
- [Traps](#), page 22
- [Management Information Base](#), page 22
- [User Datagram Protocol](#), page 23
- [Numbering system conventions](#), page 23

SNMP interface objectives

The objectives of the SNMP Interface are as follows:

- Connect to the SNMP agent that resides on the switch using a management workstation.
- Support of Fabric Element Management Information Bases (FE-MIB) (rfc2837) and Fibre Alliance Management Information Bases (FA-MIB) draft.
- Support of version 1 and 2 traps.
- The SNMP agent supports SNMPv1 and SNMPv2c.

Manager and Agent

The two primary elements of SNMP are:

- **Manager**—Application that runs on the management workstation.
- **Agent**—Daemon application that runs on the switch.

The Manager is the application through which the network administrator performs network management functions. The SNMP agent is the direct interface on the switch for any SNMP manager connecting to the switch using the SNMP protocol, as shown in [Figure 1](#). The agent is started by the script file(s) responsible for switch initialization when the switch powers up or when the switch is reset.

When an SNMP request arrives at the agent, the agent composes a message and passes it on to Switch Management to process the message and provide a response to the agent. The agent provides a response to the originator of the SNMP request. The SNMP agent does not have direct access to the internal database of the switch.

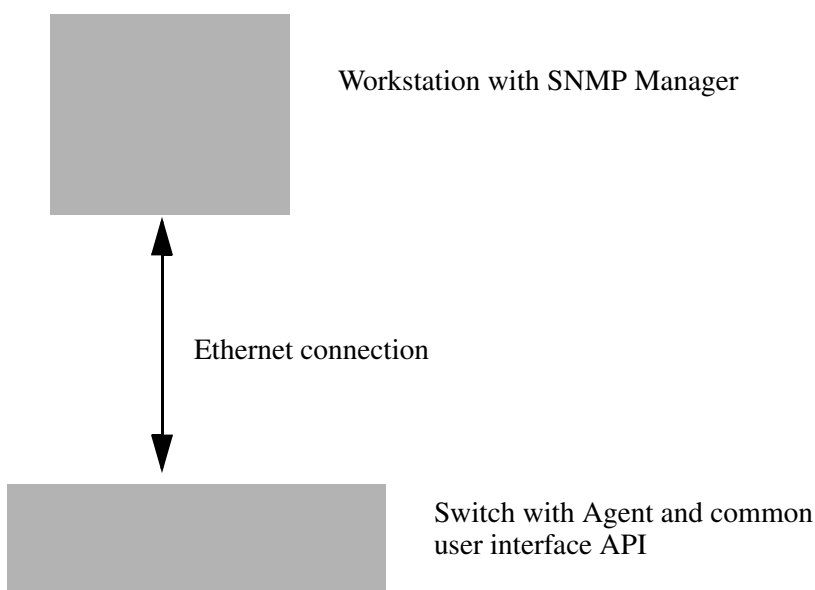


Figure 1: SNMP interface architecture

Traps

Traps are notification messages sent from the switch to a registered manager when a change of state occurs within the switch. A change of state can be an alarm condition or simply a configuration change.

The Fibre Alliance MIB defines a trap table configurable through SNMP. A trap table can have up to 5 entries, and can be configured using SNMP Manager or Switch Manager. The same trap table information is available to both SNMP Manager and Switch Manager.

A trap event is reported when the incoming error has a severity level less than or equal to the configured severity level. The trap event types and trap severity levels are listed in [Table 2](#). Refer to [Table 4, page 74](#) for information on specific traps.

Table 2: Trap severity levels

Event type	Severity level
Unknown	1
Emergency	2
Alert	3
Critical	4
Error	5
Warning	6
Notify	7
Info	8
Debug	9
Mark	10

Management Information Base

Management information bases (MIBs) define the properties of the managed object within the device to be managed. Every managed device keeps a database of values for each definition written in the MIB. It is not the actual database itself; it is implementation-dependant.

Definition of the MIB conforms to the Structure of Management Information (SMI) given in Request For Comment (RFC) 1155. The latest Internet MIB is given in RFC 1213, and is sometimes called MIB-II.

User Datagram Protocol

The 2/8q FC Switch supports the following User Datagram Protocol (UDP) settings:

- Agents “listen” on UDP port 161.
- Responses are sent back to the originating Network Management Station (NMS) port from a dynamic port, although many agents use port 161 also for this target.
- The maximum SNMP message size is 65507 octets (maximum UDP message size).
- The minimum receive packet size for SNMP implementations is 484 octets in length.
- Agent and Network Monitoring Systems are responsible for determining error recovery.

Numbering system conventions

The conventions for numbering systems in this guide are as follows:

- Decimal = 101
- Hexadecimal = 0x101
- Binary = 101b

Configuring a Switch

2

This chapter describes how to configure the 2/8q FC Switch to support SNMP. The following topics are covered:

- [System specifications and requirements](#), page 26
- [Configuring a switch using the Command Line Interface](#), page 27
- [Configuring a switch using Switch Manager](#), page 29

System specifications and requirements

- The 2/8q FC Switch supports SNMPv1 and SNMPv2c.
- Version 1 and 2 traps are supported.
- Hardware—One out-of-band Ethernet connection is required.
- Software—One switch management software application allows you to:
 - Monitor and control the switch.
 - Read, write, and receive trap information, if supported.
- Ports on the switch reserved for SNMP:
 - Port 161 is not configurable, and is used for the standard SNMP commands.
 - Port 162 is configurable and is the default port used for traps.

Configuring a switch using the Command Line Interface

The Telnet command line interface offers a convenient way to change SNMP parameters. SNMP parameter defaults are preset during manufacturing. For security purposes, these default values should be changed.

For specific information about SNMP parameters, refer to the SNMP Configuration section in the corresponding *HP StorageWorks 2/8q Fibre Channel Switch Management User Guide*. To configure a switch using the command line interface, do the following. Press the Enter key to accept the default value for each parameter.

```

cli #> admin start
cli (admin) #> set setup snmp
A list of attributes with formatting and current values will follow. Enter
a new value or simply press the ENTER key to accept the current value. If
you wish to terminate this process before reaching the end of the list
press 'q' or 'Q' and the ENTER key to do so.

Trap Severity Options
-----
unknown, emergency, alert, critical, error, warning, notify, info, debug,
mark
SnmpEnabled      (string, max=32 chars)      [True]
Contact          (string, max=32 chars)      [<sysContact undefined> ]
Location         (string, max=32 chars)      [<sysLocation undefined>]
Trap1Address     (dot-notated IP Address)    [10.0.0.254           ]
Trap1Port        (decimal value)             [162                  ]
Trap1Severity    (see allowed options above) [warning              ]
Trap1Version     (1 / 2)                     [2                    ]
Trap1Enabled     (True / False)              [False                ]
Trap2Address     (dot-notated IP Address)    [0.0.0.0              ]
Trap2Port        (decimal value)             [162                  ]
Trap2Severity    (see allowed options above) [warning              ]
Trap2Version     (1 / 2)                     [2                    ]
Trap2Enabled     (True / False)              [False                ]
Trap3Address     (dot-notated IP Address)    [0.0.0.0              ]
Trap3Port        (decimal value)             [162                  ]
Trap3Severity    (see allowed options above) [warning              ]
Trap3Version     (1 / 2)                     [2                    ]
Trap3Enabled     (True / False)              [False                ]
Trap4Address     (dot-notated IP Address)    [0.0.0.0              ]
Trap4Port        (decimal value)             [162                  ]
Trap4Severity    (see allowed options above) [warning              ]
Trap4Version     (1 / 2)                     [2                    ]
Trap4Enabled     (True / False)              [False                ]
Trap5Address     (dot-notated IP Address)    [0.0.0.0              ]
Trap5Port        (decimal value)             [162                  ]
Trap5Severity    (see allowed options above) [warning              ]
Trap5Version     (1 / 2)                     [2                    ]
Trap5Enabled     (True / False)              [False                ]
ReadCommunity    (string, max=32 chars)      [public               ]
WriteCommunity   (string, max=32 chars)      [private              ]
TrapCommunity    (string, max=32 chars)      [public               ]
AuthFailureTrap  (True / False)             [False                ]
ProxyEnabled     (True / False)             [True                 ]
Do you want to save and activate this snmp setup? (y/n) [n] y
SNMP setup saved and activated.

```

Configuring a switch using Switch Manager

To configure the 2/8q FC Switch using Switch Manager, use the Switch Properties and Network Properties windows. For specific information, refer to the corresponding *HP StorageWorks 2/8q Fibre Channel Switch Management User Guide*.

MIB-II Objects

3

This chapter covers the implementation details for the MIB-II on the 2/8q FC Switch. A MIB defines the properties of the managed object within the device to be managed. Every managed device keeps a database of values for each definition written in the MIB. It is not the actual database itself; it is implementation-dependant. Definition of the MIB conforms to the SMI given in RFC 1155. The latest Internet MIB is given in RFC 1213, and is sometimes called MIB-II.

The following topics are covered:

- [Groups in MIB-II](#), page 32
- [System group](#), page 33
- [Interfaces group](#), page 35
- [Address translation group](#), page 40
- [IP group](#), page 41
- [ICMP group](#), page 51
- [TCP group](#), page 56
- [UDP group](#), page 60
- [EGP group](#), page 62
- [Transmission group](#), page 66
- [SNMP group](#), page 67

Groups in MIB-II

Refer the [Table 3](#) for the syntax for MIB-II Groups.

Table 3: MIB-II groups

Group	Syntax
system	OBJECT IDENTIFIER ::= { mib-2 1 }
interfaces	OBJECT IDENTIFIER ::= { mib-2 2 }
at	OBJECT IDENTIFIER ::= { mib-2 3 }
ip	OBJECT IDENTIFIER ::= { mib-2 4 }
icmp	OBJECT IDENTIFIER ::= { mib-2 5 }
tcp	OBJECT IDENTIFIER ::= { mib-2 6 }
udp	OBJECT IDENTIFIER ::= { mib-2 7 }
snmp	OBJECT IDENTIFIER ::= { mib-2 11 }

System group

Implementation of the System group is mandatory for all systems. If an agent is not configured to have a value for any of these variables, a string of length 0 is returned.

sysDescr (1.3.6.1.2.1.1.1)

A textual description of the entity. This value must include the full name and version identification of the system's hardware type, operating-system, and networking software. It is mandatory that this only contain printable American Standard Code for Information Interchange (ASCII) characters.

- Syntax DisplayString—(SIZE (0..255))
- Access—Read-only
- Status—Mandatory
- Return value—Default is HP StorageWorks 2/8q FC Switch.

sysObjectID (1.3.6.1.2.1.1.2)

The vendor's authoritative identification of the network management subsystem contained in the entity. This value is allocated within the SMI enterprise subtree (1.3.6.1.4.1) and provides an easy and unambiguous means for determining what kind of box is being managed. For example, if vendor 'Flintstones, Inc.' was assigned the subtree 1.3.6.1.4.1.4242, it could assign the identifier 1.3.6.1.4.1.4242.1.1 to its 'Fred Router'.

- Syntax—OBJECT IDENTIFIER
- Access—Read-only
- Status—Mandatory
- Return value—HP StorageWorks 2/8q FC Switch = 1.3.6.1.4.1.1663.1.1.1.1.31

sysUpTime (1.3.6.1.2.1.1.3)

The time, in hundredths of a second, since the network management portion of the system was last re-initialized.

- Syntax—TimeTicks
- Access—Read-only
- Status—Mandatory
- Return value—The time since the switch was powered on, or last reset (reset, hardreset, or hotreset) was executed (for example, 3 days 21 hours, 5 minutes, and 26.84 seconds.)

sysContact (1.3.6.1.2.1.1.4)

The textual identification of the contact person for this managed node, together with information on how to contact this person.

- Syntax—DisplayString (SIZE (0..255))
- Access—Read-write
- Status—Mandatory
- Return value—The default is <sysContact undefined>. The string size is limited to a maximum of 64.

sysName (1.3.6.1.2.1.1.5)

An administratively-assigned name for this managed node. By convention, this is the node's fully-qualified domain name.

- Syntax—DisplayString (SIZE (0..255))
- Access—Read-write
- Status—Mandatory
- Return value—The default is HP StorageWorks 2/8q FC Switch.

sysLocation (1.3.6.1.2.1.1.6)

The physical location of this node, such as telephone closet and 3rd floor.

- Syntax—DisplayString (SIZE (0..255))
- Access—Read-write
- Status—Mandatory
- Return value—The default is <sysLocation undefined>. The string size is limited to a maximum of 64.

sysServices (1.3.6.1.2.1.1.7)

A value that indicates the set of services that this entity primarily offers. The value is a sum. This sum initially takes the value zero. For each layer L in the range 1 through 7 that this node performs transactions for, 2 raised to (L - 1) is added to the sum. For example, a node that performs primarily routing functions would have a value of 4 ($2^{(3-1)}$). In contrast, a node that is a host offering application services would have a value of 72 ($2^{(4-1)} + 2^{(7-1)}$).

- Syntax—INTEGER (0..127)
- Access—Read-only
- Status—Mandatory
- Return value—The default is 2.

Interfaces group

Implementation of the Interfaces group is mandatory for all systems.

ifNumber (1.3.6.1.2.1.2.1)

The number of network interfaces (regardless of their current state) present on this system.

- Syntax—INTEGER
- Access—Read-only
- Status—Mandatory
- Return value—The default is 2.

Interfaces table

The Interfaces table contains information on the entity's interfaces. Each interface is thought of as being attached to a *subnetwork*. This term must not be confused with *subnet* which refers to an addressing partitioning scheme used in the Internet suite of protocols.

ifIndex (1.3.6.1.2.1.2.2.1.1)

A unique value for each interface. Its value ranges between 1 and the value of ifNumber. The value for each interface must remain constant at least from one re-initialization of the entity's network management system to the next re-initialization.

- Syntax—INTEGER
- Access—Read-only
- Status—Mandatory

ifDescr (1.3.6.1.2.1.2.2.1.2)

A textual string containing information about the interface. This string must include the name of the manufacturer, the product name, and the version of the hardware interface.

- Syntax—DisplayString (SIZE (0..255))
- Access—Read-only
- Status—Mandatory

ifType (1.3.6.1.2.1.2.2.1.3)

The type of interface distinguished according to the physical/link protocol(s) immediately *below* the network layer in the protocol stack.

- Syntax—INTEGER
- Access—Read-only
- Status—Mandatory

ifMtu (1.3.6.1.2.1.2.2.1.4)

The size of the largest datagram which can be sent/received on the interface, specified in octets. For interfaces that are used for transmitting network datagrams, this is the size of the largest network datagram that can be sent on the interface.

- Syntax—INTEGER
- Access—Read-only
- Status—Mandatory

ifSpeed (1.3.6.1.2.1.2.2.1.5)

An estimate of the interface's current bandwidth in bits per second. For interfaces that do not vary in bandwidth or for those where no accurate estimation can be made, this object must contain the nominal bandwidth.

- Syntax—Gauge
- Access—Read-only
- Status—Mandatory

ifPhysAddress (1.3.6.1.2.1.2.2.1.6)

The interface's address at the protocol layer immediately *below* the network layer in the protocol stack. For interfaces that do not have such an address, such as a serial line, this object must contain an octet string of zero length.

- Syntax—PhysAddress
- Access—Read-only
- Status—Mandatory

ifAdminStatus (1.3.6.1.2.1.2.2.1.7)

The desired state of the interface. The testing(3) state indicates that no operational packets can be passed.

- Syntax—INTEGER
- Access—Read-write
- Status—Mandatory
- Return value—Writes not supported

ifOperStatus (1.3.6.1.2.1.2.2.1.8)

The current operational state of the interface. The testing(3) state indicates that no operational packets can be passed.

- Syntax—INTEGER
- Access—Read-only
- Status—Mandatory

ifLastChange (1.3.6.1.2.1.2.2.1.9)

The value of sysUpTime at the time the interface entered its current operational state. If the current state was entered prior to the last re-initialization of the local network management subsystem, this object contains a zero value.

- Syntax—TimeTicks
- Access—Read-only
- Status—Mandatory

ifInOctets (1.3.6.1.2.1.2.2.1.10)

The total number of octets received on the interface, including framing characters.

- Syntax—Counter
- Access—Read-only
- Status—Mandatory

ifInUcastPkts (1.3.6.1.2.1.2.2.1.11)

The number of subnetwork-unicast packets delivered to a higher-layer protocol.

- Syntax—Counter
- Access—Read-only
- Status—Mandatory

ifInNUcastPkts (1.3.6.1.2.1.2.2.1.12)

The number of non-unicast (that is, subnetwork- broadcast or subnetwork-multicast) packets delivered to a higher-layer protocol.

- Syntax—Counter
- Access—Read-only
- Status—Mandatory

ifInDiscards (1.3.6.1.2.1.2.2.1.13)

The number of inbound packets chosen to be discarded even though no errors were detected, to prevent their being deliverable to a higher-layer protocol. One possible reason for discarding such a packet is to free up buffer space.

- Syntax—Counter
- Access—Read-only
- Status—Mandatory

ifInErrors (1.3.6.1.2.1.2.2.1.14)

The number of inbound packets that contained errors, preventing them from being deliverable to a higher-layer protocol.

- Syntax—Counter
- Access—Read-only
- Status—Mandatory

ifInUnknownPkts (1.3.6.1.2.1.2.2.1.15)

The number of packets received from the interface that were discarded because of an unknown or unsupported protocol.

- Syntax—Counter
- Access—Read-only
- Status—Mandatory

ifOutOctets (1.3.6.1.2.1.2.2.1.16)

The total number of octets transmitted out of the interface, including framing characters.

- Syntax—Counter
- Access—Read-only
- Status—Mandatory

ifOutUcastPkts (1.3.6.1.2.1.2.2.1.17)

The total number of packets that higher level protocols requested be transmitted to a subnetwork unicast address, including those that were discarded or not sent.

- Syntax—Counter
- Access—Read-only
- Status—Mandatory

ifOutNUcastPkts (1.3.6.1.2.1.2.2.1.18)

The total number of packets that higher level protocols requested be transmitted to a non-unicast (subnetwork broadcast or subnetwork multicast) address, including those that were discarded or not sent.

- Syntax—Counter
- Access—Read-only
- Status—Mandatory

ifOutDiscards (1.3.6.1.2.1.2.2.1.19)

The number of outbound packets chosen to be discarded, even though no errors were detected to prevent their being transmitted. One possible reason for discarding such a packet is to free up buffer space.

- Syntax—Counter
- Access—Read-only
- Status—Mandatory

ifOutErrors (1.3.6.1.2.1.2.2.1.20)

The number of outbound packets that could not be transmitted because of errors.

- Syntax—Counter
- Access—Read-only
- Status—Mandatory

ifOutQLen (1.3.6.1.2.1.2.2.1.21)

The length (in packets) of the output packet queue.

- Syntax—Gauge
- Access—Read-only
- Status—Mandatory

ifSpecific (1.3.6.1.2.1.2.2.1.22)

A reference to MIB definitions specific to the particular media being used to realize the interface. For example, if the interface is realized by an Ethernet, the value of this object refers to a document that defines objects specific to Ethernet. If this information is not present, its value must be set to the OBJECT IDENTIFIER { 0 0 }, which is a syntactically valid object identifier, and any conformant implementation of ASN.1 (Abstract Syntax Notation) and BER must be able to generate and recognize this value.

- Syntax—OBJECT IDENTIFIER
- Access—Read-only
- Status—Mandatory

Address translation group

Implementation of the Address Translation group is mandatory for all systems. However, this group is deprecated by MIB-II. That is, it is being included solely for compatibility with MIB-I nodes, and is most likely excluded from MIB-III nodes. From MIB-II and onwards, each network protocol group contains its own address translation tables.

The Address Translation group contains one table which is the union across all interfaces of the translation tables for converting a NetworkAddress (for example, an IP address) into a subnetwork-specific address. For lack of a better term, this document refers to such a subnetwork-specific address as a 'physical' address.

Examples of such translation tables are for: broadcast media where ARP is in use, the translation table is equivalent to the ARP cache, or on an X.25 network where non-algorithmic translation to X.121 addresses is required. The translation table contains the NetworkAddress to X.121 address equivalences.

atIfIndex (1.3.6.1.2.1.3.1.1.1)

The interface on which this entry's equivalence is effective. The interface identified by a particular value of this index is the same interface as identified by the same value of ifIndex.

- Syntax—INTEGER
- Access—Read-write
- Status—Deprecated

atPhysAddress (1.3.6.1.2.1.3.1.1.2)

The media-dependent "physical" address. Setting this object to a null string (one of zero length) invalidates the corresponding entry in the atTable object. That is, it effectively disassociates the interface identified with the entry from the mapping identified with the entry. It is an implementation-specific matter as to whether the agent removes an invalidated entry from the table. Accordingly, management workstations must be prepared to receive tabular information from agents that corresponds to entries not currently in use. Proper interpretation of such entries requires examination of the relevant atPhysAddress object.

- Syntax—PhysAddress
- Access—Read-write
- Status—Deprecated

atNetAddress (1.3.6.1.2.1.3.1.1.3)

The NetworkAddress corresponding to the media-dependent 'physical' address.

- Syntax—NetworkAddress
- Access—Read-write
- Status—Deprecated

IP group

Implementation of the IP group is mandatory for all systems.

ipForwarding (1.3.6.1.2.1.4.1)

The indication of whether this entity is acting as an IP Gateway with respect to the forwarding of datagrams received by, but not addressed to, this entity. IP Gateways forward datagrams; IP hosts do not (except those source-routed from the host).

For some managed nodes, this object can take only a subset of the values possible. Accordingly, it is appropriate for an agent to return a “badValue” response if a management station attempts to change this object to an inappropriate value.

- Syntax—INTEGER
- Access—Read-write
- Status—Mandatory
- Return value—Returns forwarding (1); Writes not supported.

ipDefaultTTL (1.3.6.1.2.1.4.2)

The default value inserted into the Time-To-Live field of the IP header of datagrams originated at this entity whenever a TTL value is not supplied by the transport layer protocol.

- Syntax—INTEGER
- Access—Read-write
- Status—Mandatory
- Return value—Returns 64 (0x40). Writes not supported.

ipInReceives (1.3.6.1.2.1.4.3)

The total number of input datagrams received from interfaces, including those received in error.

- Syntax—Counter
- Access—Read-only
- Status—Mandatory

ipInHdrErrors (1.3.6.1.2.1.4.4)

The number of input datagrams discarded due to errors in their IP headers. These include bad checksums, version number mismatch, other format errors, time-to-live exceeded, and errors discovered in processing their IP options.

- Syntax—Counter
- Access—Read-only
- Status—Mandatory

ipInAddrErrors (1.3.6.1.2.1.4.5)

The number of input datagrams discarded because the IP address in their IP header's destination field was not a valid address to be received at this entity. This count includes invalid addresses (for example, 0.0.0.0) and addresses of unsupported classes (for example, Class E). For entities which are not IP Gateways and therefore do not forward datagrams, this counter includes datagrams discarded because the destination address was not a local address.

- Syntax—Counter
- Access—Read-only
- Status—Mandatory

ipForwDatagrams (1.3.6.1.2.1.4.6)

The number of input datagrams for which this entity was not their final IP destination. As a result, an attempt was made to find a route to forward them to that final destination. In entities that do not act as IP Gateways, this counter includes only those packets that were Source Routed from this entity, and the Source Route option processing was successful.

- Syntax—Counter
- Access—Read-only
- Status—Mandatory

ipInUnknownProtos (1.3.6.1.2.1.4.7)

The number of locally-addressed datagrams received successfully but discarded because of an unknown or unsupported protocol.

- Syntax—Counter
- Access—Read-only
- Status—Mandatory

ipInDiscards (1.3.6.1.2.1.4.8)

The number of input IP datagrams for which no problems were encountered to prevent their continued processing, but which were discarded (for example, for lack of buffer space). This counter does not include any datagrams discarded while awaiting reassembly.

- Syntax—Counter
- Access—Read-only
- Status—Mandatory

ipInDelivers (1.3.6.1.2.1.4.9)

The total number of input datagrams successfully delivered to IP user protocols including Internet Control Message Protocol (ICMP).

- Syntax—Counter
- Access—Read-only
- Status—Mandatory

ipOutRequests (1.3.6.1.2.1.4.10)

The total number of IP datagrams that local IP user protocols (including ICMP) supplied to IP in requests for transmission. This counter does not include any datagrams counted in ipForwDatagrams.

- Syntax—Counter
- Access—Read-only
- Status—Mandatory

ipOutDiscards (1.3.6.1.2.1.4.11)

The number of output IP datagrams for which no problem was encountered to prevent their transmission to their destination, but which were discarded (for example, for lack of buffer space). This counter would include datagrams counted in ipForwDatagrams if any such packets met this (discretionary) discard criterion.

- Syntax—Counter
- Access—Read-only
- Status—Mandatory

ipOutNoRoutes (1.3.6.1.2.1.4.12)

The number of IP datagrams discarded because no route could be found to transmit them to their destination. This counter includes any packets counted in ipForwDatagrams which meet this “no-route” criterion. This includes any datagrams that a host cannot route because all of its default gateways are down.

- Syntax—Counter
- Access—Read-only
- Status—Mandatory

ipReasmTimeout (1.3.6.1.2.1.4.13)

The maximum number of seconds which received fragments are held while they are awaiting reassembly at this entity.

- Syntax—INTEGER
- Access—Read-only
- Status—Mandatory

ipReasmReqds (1.3.6.1.2.1.4.14)

The number of IP fragments received that needed to be reassembled at this entity.

- Syntax—Counter
- Access—Read-only
- Status—Mandatory

ipReasmOKs (1.3.6.1.2.1.4.15)

The number of IP datagrams successfully reassembled.

- Syntax—Counter
- Access—Read-only
- Status—Mandatory

ipReasmFails (1.3.6.1.2.1.4.16)

The number of failures detected by the IP reassembly algorithm (for example, timed out, errors). This is not necessarily a count of discarded IP fragments, since some algorithms (notably the algorithm in RFC 815) can lose track of the number of fragments by combining them as they are received.

- Syntax—Counter
- Access—Read-only
- Status—Mandatory

ipFragOKs (1.3.6.1.2.1.4.17)

The number of IP datagrams that have been successfully fragmented at this entity.

- Syntax—Counter
- Access—Read-only
- Status—Mandatory

ipFragFails (1.3.6.1.2.1.4.18)

The number of IP datagrams that have been discarded because they needed to be fragmented at this entity, but could not because their Don't Fragment flag was set.

- Syntax—Counter
- Access—Read-only
- Status—Mandatory

ipFragCreates (1.3.6.1.2.1.4.19)

The number of IP datagram fragments that have been generated as a result of fragmentation at this entity.

- Syntax—Counter
- Access—Read-only
- Status—Mandatory

IP address table

The IP address table contains this entity's IP addressing information.

ipAdEntAddr (1.3.6.1.2.1.4.20.1.1)

The IP address to which this entry's addressing information pertains.

- Syntax—IpAddress
- Access—Read-only
- Status—Mandatory

ipAdEntIfIndex (1.3.6.1.2.1.4.20.1.2)

The index value which uniquely identifies the interface to which this entry is applicable. The interface identified by a particular value of this index is the same interface as identified by the same value of ifIndex.

- Syntax—INTEGER
- Access—Read-only
- Status—Mandatory

ipAdEntNetMask (1.3.6.1.2.1.4.20.1.3)

The subnet mask associated with the IP address of this entry. The value of the mask is an IP address with all the network bits set to 1 and all the hosts bits set to 0.

- Syntax—IpAddress
- Access—Read-only
- Status—Mandatory

ipAdEntBcastAddr (1.3.6.1.2.1.4.20.1.4)

The value of the least-significant bit in the IP broadcast address used for sending datagrams on the (logical) interface associated with the IP address of this entry. For example, when the Internet standard all-ones broadcast address is used, the value is 1. This value applies to both the subnet and network broadcasts addresses used by the entity on this (logical) interface.

- Syntax—INTEGER
- Access—Read-only
- Status—Mandatory

ipAdEntReasmMaxSize (1.3.6.1.2.1.4.20.1.5)

The size of the largest IP datagram which this entity can reassemble from incoming IP fragmented datagrams received on this interface.

- Syntax—INTEGER (0..65535)
- Access—Read-only
- Status—Mandatory

IP routing table

The IP routing table contains an entry for each route presently known to this entity.

ipRouteDest (1.3.6.1.2.1.4.21.1.1)

The destination IP address of this route. An entry with a value of 0.0.0.0 is considered a default route. Multiple routes to a single destination can appear in the table, but access to such multiple entries is dependent on the table-access mechanisms defined by the network management protocol in use.

- Syntax—IpAddress
- Access—Read-write
- Status—Mandatory
- Return value—Writes not supported

ipRouteIfIndex (1.3.6.1.2.1.4.21.1.2)

The index value which uniquely identifies the local interface through which the next hop of this route should be reached. The interface identified by a particular value of this index is the same interface as identified by the same value of ifIndex.

- Syntax—INTEGER
- Access—Read-write
- Status—Mandatory
- Return value—Writes not supported

ipRouteMetric1 (1.3.6.1.2.1.4.21.1.3)

The primary routing metric for this route. The semantics of this metric are determined by the routing protocol specified in the route's ipRouteProto value. If this metric is not used, its value must be set to -1.

- Syntax—INTEGER
- Access—Read-write
- Status—Mandatory
- Return value—Writes not supported

ipRouteMetric2 (1.3.6.1.2.1.4.21.1.4)

An alternate routing metric for this route. The semantics of this metric are determined by the routing protocol specified in the route's ipRouteProto value. If this metric is not used, its value must be set to -1.

- Syntax—INTEGER
- Access—Read-write
- Status—Mandatory
- Return value—Writes not supported

ipRouteMetric3 (1.3.6.1.2.1.4.21.1.5)

An alternate routing metric for this route. The semantics of this metric are determined by the routing protocol specified in the route's ipRouteProto value. If this metric is not used, its value must be set to -1.

- Syntax—INTEGER
- Access—Read-write
- Status—Mandatory
- Return value—Writes not supported

ipRouteMetric4 (1.3.6.1.2.1.4.21.1.6)

An alternate routing metric for this route. The semantics of this metric are determined by the routing protocol specified in the route's ipRouteProto value. If this metric is not used, its value must be set to -1.

- Syntax—INTEGER
- Access—Read-write
- Status—Mandatory
- Return value—Writes not supported

ipRouteNextHop (1.3.6.1.2.1.4.21.1.7)

The IP address of the next hop of this route. If a route is bound to an interface which is realized from a broadcast media, the value of this field is the agent's IP address on that interface.

- Syntax—IpAddress
- Access—Read-write
- Status—Mandatory
- Return value—Writes not supported

ipRouteType (1.3.6.1.2.1.4.21.1.8)

The type of route. The values direct(3) and indirect(4) refer to the notion of direct and indirect routing in the IP architecture. Setting this object to the value invalid(2) invalidates the corresponding entry in the ipRouteTable object. That is, it effectively disassociates the destination identified with the entry from the route identified with the entry. It is an implementation-specific matter whether the agent removes an invalidated entry from the table. Accordingly, management stations must be prepared to receive tabular information from agents that corresponds to entries not currently in use. Proper interpretation of such entries requires examination of the relevant ipRouteType object.

- Syntax—INTEGER
- Access—Read-write
- Status—Mandatory
- Return value—Writes not supported

ipRouteProto (1.3.6.1.2.1.4.21.1.9)

The routing mechanism through which this route was learned. Inclusion of values for gateway routing protocols is not intended to imply that hosts should support those protocols.

- Syntax—INTEGER
- Access—Read-write
- Status—Mandatory

ipRouteAge (1.3.6.1.2.1.4.21.1.10)

The number of seconds since this route was last updated or otherwise determined to be correct. No semantics of *too old* can be implied except through knowledge of the routing protocol by which the route was learned.

- Syntax—INTEGER
- Access—Read-write
- Status—Mandatory
- Return value—Writes not supported

ipRouteMask (1.3.6.1.2.1.4.21.1.11)

Indicate the mask to be logical-ANDed with the destination address before being compared to the value in the ipRouteDest field.

- Syntax—IpAddress
- Access—Read-write
- Status—Mandatory
- Return value—Writes not supported

ipRouteMetric5 (1.3.6.1.2.1.4.21.1.12)

An alternate routing metric for this route. The semantics of this metric are determined by the routing-protocol specified in the route's ipRouteProto value. If this metric is not used, its value must be set to -1.

- Syntax—INTEGER
- Access—Read-write
- Status—Mandatory
- Return value—Writes not supported

ipRouteInfo (1.3.6.1.2.1.4.21.1.13)

A reference to MIB definitions specific to the particular routing protocol which is responsible for this route, as determined by the value specified in the route's ipRouteProto value. If this information is not present, its value must be set to the OBJECT IDENTIFIER { 0 0 }, which is a syntactically valid object identifier. Any conformant implementation of ASN.1 and BER must be able to generate and recognize this value.

- Syntax—OBJECT IDENTIFIER
- Access—Read-only
- Status—Mandatory

IP address translation table

The IP address translation table contain the IpAddress to *physical* address equivalences. Some interfaces do not use translation tables for determining address equivalences (for example, DDN-X.25 has an algorithmic method). If all interfaces are of this type, the Address Translation table is empty; that is, it has zero entries.

ipNetToMediaIfIndex (1.3.6.1.2.1.4.22.1.1)

The interface on which this entry's equivalence is effective. The interface identified by a particular value of this index is the same interface as identified by the same value of ifIndex.

- Syntax—INTEGER
- Access—Read-write
- Status—Mandatory
- Return value—Writes not supported

ipNetToMediaPhysAddress (1.3.6.1.2.1.4.22.1.2)

The media-dependent *physical* address.

- Syntax—PhysAddress
- Access—Read-write
- Status—Mandatory
- Return value—Writes not supported

ipNetToMediaNetAddress (1.3.6.1.2.1.4.22.1.3)

The IpAddress corresponding to the media-dependent *physical* address.

- Syntax—IpAddress
- Access—Read-write
- Status—Mandatory
- Return value—Writes not supported

ipNetToMediaType (1.3.6.1.2.1.4.22.1.4)

The type of mapping. Setting this object to the value invalid(2) invalidates the corresponding entry in the ipNetToMediaTable. That is, it effectively disassociates the interface identified with the entry from the mapping identified with the entry. It is an implementation-specific matter whether the agent removes an invalidated entry from the table. Accordingly, management stations must be prepared to receive tabular information from agents that corresponds to entries not currently in use. Proper interpretation of such entries requires examination of the relevant ipNetToMediaType object.

- Syntax—INTEGER
- Access—Read-write
- Status—Mandatory
- Return value—Writes not supported

Additional IP objects

Following are the additional IP objects.

ipRoutingDiscards (1.3.6.1.2.1.4.23)

The number of routing entries chosen to be discarded even though they are valid. One possible reason for discarding such an entry is to free-up buffer space for other routing entries.

- Syntax—Counter
- Access—Read-only
- Status—Mandatory

ICMP group

Implementation of the ICMP group is mandatory for all systems.

icmpInMsgs (1.3.6.1.2.1.5.1)

The total number of ICMP messages received by the entity. This counter includes all those counted by icmpInErrors.

- Syntax—Counter
- Access—Read-only
- Status—Mandatory

icmpInErrors (1.3.6.1.2.1.5.2)

The number of ICMP messages received by the entity that were determined to have ICMP-specific errors (such as, bad ICMP checksums, bad length).

- Syntax—Counter
- Access—Read-only
- Status—Mandatory

icmpInDestUnreachs (1.3.6.1.2.1.5.3)

The number of ICMP Destination Unreachable messages received.

- Syntax—Counter
- Access—Read-only
- Status—Mandatory

icmpInTimeExcds (1.3.6.1.2.1.5.4)

The number of ICMP Time Exceeded messages received.

- Syntax—Counter
- Access—Read-only
- Status—Mandatory

icmpInParmProbs (1.3.6.1.2.1.5.5)

The number of ICMP Parameter Problem messages received.

- Syntax—Counter
- Access—Read-only
- Status—Mandatory

icmpInSrcQuenchs (1.3.6.1.2.1.5.6)

The number of ICMP Source Quench messages received.

- Syntax—Counter
- Access—Read-only
- Status—Mandatory

icmpInRedirects (1.3.6.1.2.1.5.7)

The number of ICMP Redirect messages received.

- Syntax—Counter
- Access—Read-only
- Status—Mandatory

icmpInEchos (1.3.6.1.2.1.5.8)

The number of ICMP Echo (request) messages received.

- Syntax—Counter
- Access—Read-only
- Status—Mandatory

icmpInEchoReps (1.3.6.1.2.1.5.9)

The number of ICMP Echo Reply messages received.

- Syntax—Counter
- Access—Read-only
- Status—Mandatory

icmpInTimestamps (1.3.6.1.2.1.5.10)

The number of ICMP Timestamp (request) messages received.

- Syntax—Counter
- Access—Read-only
- Status—Mandatory

icmpInTimestampReps (1.3.6.1.2.1.5.11)

The number of ICMP Timestamp Reply messages received.

- Syntax—Counter
- Access—Read-only
- Status—Mandatory

icmpInAddrMasks (1.3.6.1.2.1.5.12)

The number of ICMP Address Mask Request messages received.

- Syntax—Counter
- Access—Read-only
- Status—Mandatory

icmpInAddrMaskReps (1.3.6.1.2.1.5.13)

The number of ICMP Address Mask Reply messages received.

- Syntax—Counter
- Access—Read-only
- Status—Mandatory

icmpOutMsgs (1.3.6.1.2.1.5.14)

The total number of ICMP messages this entity attempted to send. This counter includes all those counted by icmpOutErrors.

- Syntax—Counter
- Access—Read-only
- Status—Mandatory

icmpOutErrors (1.3.6.1.2.1.5.15)

The number of ICMP messages this entity did not send due to problems discovered within ICMP, such as a lack of buffers. This value must not include errors discovered outside the ICMP layer, such as the inability of IP to route the resultant datagram. In some implementations, there can be no types of errors which contribute to this counter's value.

- Syntax—Counter
- Access—Read-only
- Status—Mandatory

icmpOutDestUnreachs (1.3.6.1.2.1.5.16)

The number of ICMP Destination Unreachable messages sent.

- Syntax—Counter
- Access—Read-only
- Status—Mandatory

icmpOutTimeExcds (1.3.6.1.2.1.5.17)

The number of ICMP Time Exceeded messages sent.

- Syntax—Counter
- Access—Read-only
- Status—Mandatory

icmpOutParmProbs (1.3.6.1.2.1.5.18)

The number of ICMP Parameter Problem messages sent.

- Syntax—Counter
- Access—Read-only
- Status—Mandatory

icmpOutSrcQuenchs (1.3.6.1.2.1.5.19)

The number of ICMP Source Quench messages sent.

- Syntax—Counter
- Access—Read-only
- Status—Mandatory

icmpOutRedirects (1.3.6.1.2.1.5.20)

The number of ICMP Redirect messages sent. For a host, this object is always zero, since hosts do not send redirects.

- Syntax—Counter
- Access—Read-only
- Status—Mandatory

icmpOutEchos (1.3.6.1.2.1.5.21)

The number of ICMP Echo (request) messages sent.

- Syntax—Counter
- Access—Read-only
- Status—Mandatory

icmpOutEchoReps (1.3.6.1.2.1.5.22)

The number of ICMP Echo Reply messages sent.

- Syntax—Counter
- Access—Read-only
- Status—Mandatory

icmpOutTimestamps (1.3.6.1.2.1.5.23)

The number of ICMP Timestamp (request) messages sent.

- Syntax—Counter
- Access—Read-only
- Status—Mandatory

icmpOutTimestampReps (1.3.6.1.2.1.5.24)

The number of ICMP Timestamp Reply messages sent.

- Syntax—Counter
- Access—Read-only
- Status—Mandatory

icmpOutAddrMasks (1.3.6.1.2.1.5.25)

The number of ICMP Address Mask Request messages sent.

- Syntax—Counter
- Access—Read-only
- Status—Mandatory

icmpOutAddrMaskReps (1.3.6.1.2.1.5.26)

The number of ICMP Address Mask Reply messages sent.

- Syntax—Counter
- Access—Read-only
- Status—Mandatory

TCP group

Implementation of the TCP group is mandatory for all systems that implement the TCP. Instances of object types that represent information about a particular TCP connection are transient; they persist only as long as the connection in question.

tcpRtoAlgorithm (1.3.6.1.2.1.6.1)

The algorithm used to determine the timeout value used for retransmitting unacknowledged octets.

- Syntax—INTEGER
- Access—Read-only
- Status—Mandatory

tcpRtoMin (1.3.6.1.2.1.6.2)

The minimum value permitted by a TCP implementation for the retransmission timeout, measured in milliseconds. More refined semantics for objects of this type depend upon the algorithm used to determine the retransmission timeout. In particular, when the timeout algorithm is rsre(3), an object of this type has the semantics of the LBOUND quantity described in RFC 793.

- Syntax—INTEGER
- Access—Read-only
- Status—Mandatory

tcpRtoMax (1.3.6.1.2.1.6.3)

The maximum value permitted by a TCP implementation for the retransmission timeout, measured in milliseconds. More refined semantics for objects of this type depend upon the algorithm used to determine the retransmission timeout. In particular, when the timeout algorithm is rsre(3), an object of this type has the semantics of the UBOUND quantity described in RFC 793.

- Syntax—INTEGER
- Access—Read-only
- Status—Mandatory

tcpMaxConn (1.3.6.1.2.1.6.4)

The maximum number of TCP connections the entity can support. In entities where the maximum number of connections is dynamic, this object must contain the value -1.

- Syntax—INTEGER
- Access—Read-only
- Status—Mandatory

tcpActiveOpens (1.3.6.1.2.1.6.5)

The number of times TCP connections have made a direct transition to the SYN-SENT state from the CLOSED state.

- Syntax—Counter
- Access—Read-only
- Status—Mandatory

tcpPassiveOpens (1.3.6.1.2.1.6.6)

The number of times TCP connections have made a direct transition to the SYN-RCVD state from the LISTEN state.

- Syntax—Counter
- Access—Read-only
- Status—Mandatory

tcpAttemptFails (1.3.6.1.2.1.6.7)

The number of times TCP connections have made a direct transition to the CLOSED state from either the SYN-SENT state or the SYN-RCVD state, plus the number of times TCP connections have made a direct transition to the LISTEN state from the SYN-RCVD state.

- Syntax—Counter
- Access—Read-only
- Status—Mandatory

tcpEstabResets (1.3.6.1.2.1.6.8)

The number of times TCP connections have made a direct transition to the CLOSED state from either the ESTABLISHED state or the CLOSE-WAIT state.

- Syntax—Counter
- Access—Read-only
- Status—Mandatory

tcpCurrEstab (1.3.6.1.2.1.6.9)

The number of TCP connections for which the current state is either ESTABLISHED or CLOSE-WAIT.

- Syntax—Gauge
- Access—Read-only
- Status—Mandatory

tcpInSegs (1.3.6.1.2.1.6.10)

The total number of segments received, including those received in error. This count includes segments received on currently established connections.

- Syntax—Counter
- Access—Read-only
- Status—Mandatory

tcpOutSegs (1.3.6.1.2.1.6.11)

The total number of segments sent including those on current connections, but excluding those containing only retransmitted octets.

- Syntax—Counter
- Access—Read-only
- Status—Mandatory

tcpRetransSegs (1.3.6.1.2.1.6.12)

The total number of segments retransmitted. That is, the number of TCP segments transmitted containing one or more previously transmitted octets.

- Syntax—Counter
- Access—Read-only
- Status—Mandatory

TCP connection table

The TCP connection table contains information about this entity's existing TCP connections.

tcpConnState (1.3.6.1.2.1.6.13.1.1)

The state of this TCP connection. The only value which can be set by a management station is deleteTCB(12). Accordingly, it is appropriate for an agent to return a “badValue” response if a management station attempts to set this object to any other value.

If a management station sets this object to the value deleteTCB(12), this deletes the TCB (as defined in RFC 793) of the corresponding connection on the managed node. The result is an immediate termination of the connection.

- Syntax—INTEGER
- Access—Read-only
- Status—Mandatory
- Return value—Writes not supported

tcpConnLocalAddress (1.3.6.1.2.1.6.13.1.2)

The local IP address for this TCP connection. If a connection is in the LISTEN state which is willing to accept connections for any IP interface associated with the node, the value 0.0.0.0 is used.

- Syntax—IpAddress
- Access—Read-only
- Status—Mandatory

tcpConnLocalPort (1.3.6.1.2.1.6.13.1.3)

The local port number for this TCP connection.

- Syntax—INTEGER (0..65535)
- Access—Read-only
- Status—Mandatory

tcpConnRemAddress (1.3.6.1.2.1.6.13.1.4)

The remote IP address for this TCP connection.

- Syntax—IpAddress
- Access—Read-only
- Status—Mandatory

tcpConnRemPort (1.3.6.1.2.1.6.13.1.5)

The remote port number for this TCP connection.

- Syntax—INTEGER (0..65535)
- Access—Read-only
- Status—Mandatory

Additional TCP objects

Following are the additional TCP objects.

tcpInErrs (1.3.6.1.2.1.6.14)

The total number of segments received in error (for example, bad TCP checksums).

- Syntax—Counter
- Access—Read-only
- Status—Mandatory

tcpOutRsts (1.3.6.1.2.1.6.15)

The number of TCP segments sent containing the RST flag.

- Syntax—Counter
- Access—Read-only
- Status—Mandatory

UDP group

Implementation of the UDP group is mandatory for all systems which implement the UDP.

udpInDatagrams (1.3.6.1.2.1.7.1)

The total number of UDP datagrams delivered to UDP users.

- Syntax—Counter
- Access—Read-only
- Status—Mandatory

udpNoPorts (1.3.6.1.2.1.7.2)

The total number of received UDP datagrams for which there was no application at the destination port.

- Syntax—Counter
- Access—Read-only
- Status—Mandatory

udpInErrors (1.3.6.1.2.1.7.3)

The number of received UDP datagrams that could not be delivered for reasons other than the lack of an application at the destination port.

- Syntax—Counter
- Access—Read-only
- Status—Mandatory

udpOutDatagrams (1.3.6.1.2.1.7.4)

The total number of UDP datagrams sent from this entity.

- Syntax—Counter
- Access—Read-only
- Status—Mandatory

UDP listener table

The UDP listener table contains information about this entity's UDP end-points on which a local application is currently accepting datagrams.

udpLocalAddress (1.3.6.1.2.1.7.5.1.1)

The local IP address for this UDP listener. In the case of a UDP listener which is willing to accept datagrams for any IP interface associated with the node, the value 0.0.0.0 is used.

- Syntax—IpAddress
- Access—Read-only
- Status—Mandatory

udpLocalPort (1.3.6.1.2.1.7.5.1.2)

The local port number for this UDP listener.

- Syntax—INTEGER (0..65535)
- Access—Read-only
- Status—Mandatory

EGP group

Implementation of the Exterior Gateway Protocol (EGP) group is mandatory for all systems which implement the EGP.

egpInMsgs (1.3.6.1.2.1.8.1)

The number of EGP messages received without error.

- Syntax—Counter
- Access—Read-only
- Status—Mandatory

egpInErrors (1.3.6.1.2.1.8.2)

The number of EGP messages received that proved to be in error.

- Syntax—Counter
- Access—Read-only
- Status—Mandatory

egpOutMsgs (1.3.6.1.2.1.8.3)

The total number of locally generated EGP messages.

- Syntax—Counter
- Access—Read-only
- Status—Mandatory

egpOutErrors (1.3.6.1.2.1.8.4)

The number of locally generated EGP messages not sent due to resource limitations within an EGP entity.

- Syntax—Counter
- Access—Read-only
- Status—Mandatory

EGP neighbor table

The EGP neighbor table contains information about this entity's EGP neighbors.

egpNeighState (1.3.6.1.2.1.8.5.1.1)

The EGP state of the local system with respect to this entry's EGP neighbor. Each EGP state is represented by a value that is one greater than the numerical value associated with the state in RFC 904.

- Syntax—INTEGER
- Access—Read-only
- Status—Mandatory

egpNeighAddr (1.3.6.1.2.1.8.5.1.2)

The IP address of this entry's EGP neighbor.

- Syntax—IpAddress
- Access—Read-only
- Status—Mandatory

egpNeighAs (1.3.6.1.2.1.8.5.1.3)

The autonomous system of this EGP peer. Zero must be specified if the autonomous system number of the neighbor is not yet known.

- Syntax—INTEGER
- Access—Read-only
- Status—Mandatory

egpNeighInMsgs (1.3.6.1.2.1.8.5.1.4)

The number of EGP messages received without error from this EGP peer.

- Syntax—Counter
- Access—Read-only
- Status—Mandatory

egpNeighInErrs (1.3.6.1.2.1.8.5.1.5)

The number of EGP messages received from this EGP peer in error (for example, bad EGP checksum).

- Syntax—Counter
- Access—Read-only
- Status—Mandatory

egpNeighOutMsgs (1.3.6.1.2.1.8.5.1.6)

The number of locally generated EGP messages to this EGP peer.

- Syntax—Counter
- Access—Read-only
- Status—Mandatory

egpNeighOutErrs (1.3.6.1.2.1.8.5.1.7)

The number of locally generated EGP messages not sent to this EGP peer due to resource limitations within an EGP entity.

- Syntax—Counter
- Access—Read-only
- Status—Mandatory

egpNeighInErrMsgs (1.3.6.1.2.1.8.5.1.8)

The number of EGP-defined error messages received from this EGP peer.

- Syntax—Counter
- Access—Read-only
- Status—Mandatory

egpNeighOutErrMsgs (1.3.6.1.2.1.8.5.1.9)

The number of EGP-defined error messages sent to this EGP peer.

- Syntax—Counter
- Access—Read-only
- Status—Mandatory

egpNeighStateUps (1.3.6.1.2.1.8.5.1.10)

The number of EGP state transitions to the UP state with this EGP peer.

- Syntax—Counter
- Access—Read-only
- Status—Mandatory

egpNeighStateDowns (1.3.6.1.2.1.8.5.1.11)

The number of EGP state transitions from the UP state to any other state with this EGP peer.

- Syntax—Counter
- Access—Read-only
- Status—Mandatory

egpNeighIntervalHello (1.3.6.1.2.1.8.5.1.12)

The interval between EGP Hello command retransmissions, in hundredths of a second. This represents the t1 timer as defined in RFC 904.

- Syntax—INTEGER
- Access—Read-only
- Status—Mandatory

egpNeighIntervalPoll (1.3.6.1.2.1.8.5.1.13)

The interval between EGP poll command retransmissions, in hundredths of a second. This represents the t3 timer as defined in RFC 904.

- Syntax—INTEGER
- Access—Read-only
- Status—Mandatory

egpNeighMode (1.3.6.1.2.1.8.5.1.14)

The polling mode of this EGP entity, either passive or active.

- Syntax—INTEGER { active(1); passive(2) }
- Access—Read-only
- Status—Mandatory

egpNeighEventTrigger (1.3.6.1.2.1.8.5.1.15)

A control variable used to trigger operator-initiated Start and Stop events. When read, this variable always returns the most recent value to which that `egpNeighEventTrigger` was set. If it has not been set since the last initialization of the network management subsystem on the node, it returns a value of “stop”.

When set, this variable causes a Start or Stop event on the specified neighbor, as specified on pages 8-10 of RFC 904. Briefly, a Start event causes an Idle peer to begin neighbor acquisition and a non-Idle peer to re-initiate neighbor acquisition. A Stop event causes a non-Idle peer to return to the Idle state until a Start event occurs, either by `egpNeighEventTrigger` or otherwise.

- Syntax—INTEGER { start(1); stop(2) }
- Access—Read-write
- Status—Mandatory

egpAs (1.3.6.1.2.1.8.6)

The autonomous system number of this EGP entity.

- Syntax—INTEGER
- Access—Read-only
- Status—Mandatory

Transmission group

Based on the transmission media underlying each interface on a system, the corresponding portion of the Transmission group is mandatory for that system.

When Internet-standard definitions for managing transmission media are defined, the transmission group is used to provide a prefix for the names of those objects.

Typically, such definitions reside in the experimental portion of the MIB until they are "proven", then as a part of the Internet standardization process, the definitions are accordingly elevated and a new object identifier, under the transmission group is defined. By convention, the name assigned is:

type OBJECT IDENTIFIER ::= { transmission number }.

Where "type" is the symbolic value used for the media in the ifType column of the ifTable object, and "number" is the actual integer value corresponding to the symbol.

SNMP group

Implementation of the SNMP group is mandatory for all systems which support an SNMP protocol entity. Some of the objects defined below are zero-valued in those SNMP implementations that are optimized to support only those functions specific to either a management agent or a management station. Objects below refer to an SNMP entity, and there can be several SNMP entities residing on a managed node.

snmplnPks (1.3.6.1.2.1.11.1)

The total number of messages delivered to the SNMP entity from the transport service.

- Syntax—Counter
- Access—Read-only
- Status—Mandatory

snmpOutPkts (1.3.6.1.2.1.11.2)

The total number of SNMP messages passed from the SNMP protocol entity to the transport service.

- Syntax—Counter
- Access—Read-only
- Status—Mandatory

snmplnBadVersions (1.3.6.1.2.1.11.3)

The total number of SNMP messages delivered to the SNMP protocol entity and intended for an unsupported SNMP version.

- Syntax—Counter
- Access—Read-only
- Status—Mandatory

snmplnBadCommunityNames (1.3.6.1.2.1.11.4)

The total number of SNMP messages delivered to the SNMP protocol entity that used a SNMP community name not known to the entity.

- Syntax—Counter
- Access—Read-only
- Status—Mandatory

snmplnBadCommunityUses (1.3.6.1.2.1.11.5)

The total number of SNMP messages delivered to the SNMP protocol entity which represented an SNMP operation which was not allowed by the SNMP community named in the message.

- Syntax—Counter
- Access—Read-only
- Status—Mandatory

snmplnASNParseErrs (1.3.6.1.2.1.11.6)

The total number of ASN.1 or BER errors encountered by the SNMP protocol entity when decoding received SNMP messages.

- Syntax—Counter
- Access—Read-only
- Status—Mandatory

snmplnTooBig (1.3.6.1.2.1.11.8)

The total number of SNMP PDUs delivered to the SNMP protocol entity for which the value of the error-status field is “tooBig”.

- Syntax—Counter
- Access—Read-only
- Status—Mandatory

snmplnNoSuchNames (1.3.6.1.2.1.11.9)

The total number of SNMP PDUs delivered to the SNMP protocol entity for which the value of the error-status field is “NoSuchName”.

- Syntax—Counter
- Access—Read-only
- Status—Mandatory

snmplnBadValues (1.3.6.1.2.1.11.10)

The total number of SNMP PDUs delivered to the SNMP protocol entity for which the value of the error-status field is “badValue”.

- Syntax—Counter
- Access—Read-only
- Status—Mandatory

snmplnReadOnly (1.3.6.1.2.1.11.11)

The total number valid SNMP PDUs delivered to the SNMP protocol entity for which the value of the error-status field is “readOnly”. It is a protocol error to generate an SNMP PDU which contains the value “readOnly” in the error-status field; as such, this object detects incorrect implementations of the SNMP.

- Syntax—Counter
- Access—Read-only
- Status—Mandatory

snmplnGenErrs (1.3.6.1.2.1.11.12)

The total number of SNMP PDUs delivered to the SNMP protocol entity for which the value of the error-status field is “genErr”.

- Syntax—Counter
- Access—Read-only
- Status—Mandatory

snmplnTotalReqVars (1.3.6.1.2.1.11.13)

The total number of MIB objects retrieved successfully by the SNMP protocol entity as the result of receiving valid SNMP Get-Request and Get-Next PDUs.

- Syntax—Counter
- Access—Read-only
- Status—Mandatory

snmplnTotalSetVars (1.3.6.1.2.1.11.14)

The total number of MIB objects altered successfully by the SNMP protocol entity as the result of receiving valid SNMP Set-Request PDUs.

- Syntax—Counter
- Access—Read-only
- Status—Mandatory

snmplnGetRequests (1.3.6.1.2.1.11.15)

The total number of SNMP Get-Request PDUs accepted and processed by the SNMP protocol entity.

- Syntax—Counter
- Access—Read-only
- Status—Mandatory

snmplnGetNexts (1.3.6.1.2.1.11.16)

The total number of SNMP Get-Next PDUs accepted and processed by the SNMP protocol entity.

- Syntax—Counter
- Access—Read-only
- Status—Mandatory

snmplnSetRequests (1.3.6.1.2.1.11.17)

The total number of SNMP Set-Request PDUs accepted and processed by the SNMP protocol entity.

- Syntax—Counter
- Access—Read-only
- Status—Mandatory

snmplnGetResponses (1.3.6.1.2.1.11.18)

The total number of SNMP Get-Response PDUs accepted and processed by the SNMP protocol entity.

- Syntax—Counter
- Access—Read-only
- Status—Mandatory

snmplnTraps (1.3.6.1.2.1.11.19)

The total number of SNMP Trap PDUs accepted and processed by the SNMP protocol entity.

- Syntax—Counter
- Access—Read-only
- Status—Mandatory

snmpOutTooBigs (1.3.6.1.2.1.11.20)

The total number of SNMP PDUs generated by the SNMP protocol entity for which the value of the error-status field is “tooBig”

- Syntax—Counter
- Access—Read-only
- Status—Mandatory

snmpOutNoSuchNames (1.3.6.1.2.1.11.21)

The total number of SNMP PDUs generated by the SNMP protocol entity for which the value of the error-status is "NoSuchName".

- Syntax—Counter
- Access—Read-only
- Status—Mandatory

snmpOutBadValues (1.3.6.1.2.1.11.22)

The total number of SNMP PDUs generated by the SNMP protocol entity for which the value of the error-status field is "badValue."

- Syntax—Counter
- Access—Read-only
- Status—Mandatory

snmpOutGenErrs (1.3.6.1.2.1.11.24)

The total number of SNMP PDUs generated by the SNMP protocol entity and for which the value of the error-status field is "genErr."

- Syntax—Counter
- Access—Read-only
- Status—Mandatory

snmpOutGetRequests (1.3.6.1.2.1.11.25)

The total number of SNMP Get-Request PDUs generated by the SNMP protocol entity.

- Syntax—Counter
- Access—Read-only
- Status—Mandatory

snmpOutGetNexts (1.3.6.1.2.1.11.26)

The total number of SNMP Get-Next PDUs generated by the SNMP protocol entity.

- Syntax—Counter
- Access—Read-only
- Status—Mandatory

snmpOutSetRequests (1.3.6.1.2.1.11.27)

The total number of SNMP Set-Request PDUs generated by the SNMP protocol entity.

- Syntax—Counter
- Access—Read-only
- Status—Mandatory

snmpOutGetResponses (1.3.6.1.2.1.11.28)

The total number of SNMP Get-Response PDUs generated by the SNMP protocol entity.

- Syntax—Counter
- Access—Read-only
- Status—Mandatory

snmpOutTraps (1.3.6.1.2.1.11.29)

The total number of SNMP Trap PDUs generated by the SNMP protocol entity.

- Syntax—Counter
- Access—Read-only
- Status—Mandatory

snmpEnableAuthenTraps (1.3.6.1.2.1.11.30)

Indicates whether the SNMP agent process is permitted to generate authentication-failure traps. The value of this object overrides any configuration information; as such, it disables all authentication-failure traps.

HP recommends that this object be stored in non-volatile memory so that it remains constant between re-initializations of the network management system.

- Syntax—INTEGER { enabled(1); disabled(2) }
- Access—Read-only
- Status—Mandatory
- Return value—Read returns enabled (1) if AuthFailureTrap = True; otherwise disabled (2). Writes not supported.

Fibre Alliance MIB Objects

4

This chapter covers the implementation details for the Fibre Alliance Management Information Bases (FA-MIB) version 4.0 on the 2/8q FC Switch.

The following topics are covered:

- [FA MIB definitions](#), page 74
- [Connectivity unit group](#), page 76
- [Connectivity table](#), page 78
- [Revision table](#), page 87
- [Sensor table](#), page 89
- [Port table](#), page 93
- [Event table](#), page 102
- [Link table](#), page 106
- [Zone table](#), page 110
- [Zoning alias table](#), page 113
- [Port statistics table](#), page 115
- [Simple Name Server \(SNS\) table](#), page 127
- [Platform table](#), page 131
- [Trap table](#), page 134
- [Related traps](#), page 137

FA MIB definitions

The FA-MIB version 4.0 is a collection of structured objects that resides on the workstation with the manager application. These objects define the syntax for information exchanged between the manager and the agent. The textual substitutions in [Table 4](#) are specific to the FA-MIB and can be used in place of primitive data types.

Table 4: FA-MIB textual substitutions

Description	Syntax
FcNameId	OCTET STRING (SIZE(8))
FcGlobalId	OCTET STRING (SIZE(16))
FcAddressId	OCTET STRING (SIZE(3))
FcEventSeverity	INTEGER{ unknown (1), emergency (2), alert (3), critical (4), error (5), warning (6), notify (7), info (8), debug (9), mark (10) - All messages logged }
FcUnitType	INTEGER { unknown(1) other(2) - none of the following hub(3) - passive connectivity unit supporting loop protocol. switch(4) - active connectivity unit supporting multiple protocols. gateway(5) - unit that converts not only the interface but also encapsulates the frame into another protocol. The assumption is that there is always two gateways connected together. For example, FC <-> ATM. converter(6) - unit that converts from one interface to another. For example, FC <-> SCSI. hba(7) - host bus adapter proxy-agent(8) - software proxy-agent storage-device(9) - disk, cd, tape, etc. host(10) - host computer storage-subsystem(11) - raid, library, etc. module(12) - subcomponent of a system swdriver(13) - software driver storage-access-device(14) - Provides storage management and access for heterogeneous hosts and heterogeneous devices wdm(15) - waveform division multiplexer ups(16) - uninterruptable power supply }

revisionNumber

The revision number for this MIB. The format of the revision value is as follows:

- (0) = high order major revision number
- (1) = low order major revision number
- (2) = high order minor revision number
- (3) = low order minor revision number

The value is stored as an ASCII value. The following is the current value of 04.00 for this object.

- (0) = '0'
 - (1) = '4'
 - (2) = '0'
 - (3) = '0'
- Syntax—DisplayString (SIZE (4))
 - Access—Read-only
 - Status—Mandatory
 - Return value—Four-digit ASCII value (for example, 0400 for MIB revision 4.0)

Connectivity unit group

The objects described in this section are not in a table format. An example of how to access one of these objects is:

```
"snmpget localhost public fcmgmt.connSet.uNumber.0".
```

uNumber (1.3.6.1.3.94.1.1)

The number of connectivity units present on this system (represented by this agent). Can be a count of the boards in a chassis or the number of full boxes in a rack.

- Syntax—INTEGER
- Access—Read-only
- Status—Mandatory
- Return value—Number of switches in the fabric

systemURL (1.3.6.1.3.94.1.2)

The top-level URL of the system. If it does not exist, the value is an empty string. The URL format is implementation dependant and can have keywords embedded that are preceded by a percent sign (for example, %USER).

- Syntax—DisplayString
- Access—Read-write
- Status—Mandatory
- Return value—Switch IP address (for example, http://10.0.0.1.). Writes not supported, returns 'NoSuchName'.

statusChangeTime (1.3.6.1.3.94.1.3)

The sysuptime timestamp at which the last status change occurred for any members of the set, in centiseconds.

- Syntax—TimeTicks
- Access—Read-only
- Status—Obsolete
- Return value—Object is obsolete; always returns error status "NoSuchName"

configurationChangeTime (1.3.6.1.3.94.1.4)

The sysuptime timestamp the last configuration change occurred for any members of the set, in centiseconds. This represents a union of change information for connUnitConfigurationChangeTime.

- Syntax—TimeTicks
- Access—Read-only
- Status—Obsolete
- Return value—Object is obsolete; always returns error status "NoSuchName"

connUnitTableChangeTime (1.3.6.1.3.94.1.5)

The sysuptime timestamp the connUnitTable was updated (an entry was either added or deleted), in centiseconds.

- Syntax—TimeTicks
- Access—Read-only
- Status—Obsolete
- Return value—Object is obsolete; always returns error status "NoSuchName"

Connectivity table

The objects described in this section are in a table format indexed by switch World Wide Name. An example of how to access one of these objects given a WWN of 100000c0dd0090a7 is:

```
"snmpget localhost public
fcmgmt.connSet.connUnitTable.connUnitEntry.connUnitId..16.0.0.192.221.0.1
44.167.0.0.0.0.0.0.0".
```

connUnitId (1.3.6.1.3.94.1.6.1.1)

The unique identification for this connectivity unit within this proxy domain. The value must be unique because it is the index variable for connUnitTable. The value assigned to a given connectivity unit must be persistent across agent and unit resets. It must be the same as connUnitGlobalId if connUnitGlobalId is known and stable.

- Syntax—FcGlobalId
- Access—Read-only
- Status—Mandatory
- Return value—World Wide Name of the switch followed by 8 bytes of zeros (for example: 10 00 00 C0 DD 00 71 C9 00 00 00 00 00 00 00).

connUnitGlobalId (1.3.6.1.3.94.1.6.1.2)

An optional global-scope identifier for this connectivity unit. It must be a WWN for this connectivity unit or 16 octets of value zero.

- Syntax—connUnitGlobalId
- Access—Read-only
- Status—Mandatory
- Return values—World Wide Name of the switch followed by 8 bytes of zeros (for example: 10 00 00 C0 DD 00 71 C9 00 00 00 00 00 00 00 00).

connUnitType (1.3.6.1.3.94.1.6.1.3)

The type of this connectivity unit.

- Syntax—FcUnitType
- Access—Read-only
- Status—Mandatory
- Return value—Switch (4)

connUnitNumports (1.3.6.1.3.94.1.6.1.4)

Number of physical ports in the connectivity unit (internal/embedded, external).

- Syntax—INTEGER
- Access—Read-only
- Status—Mandatory
- Return value—Number of ports on the switch

connUnitState (1.3.6.1.3.94.1.6.1.5)

The operational state of the switch mapped. The overall state of connectivity unit.

- Syntax—INTEGER
- Access—Read-only
- Status—Mandatory
- Return value—Refer to [Table 5](#) for switch operational states.

Table 5: Switch operational states

Switch state	Return state
online	online (2)
offline	offline (3)
diagnostics	offline (3)
other	unknown (1)

connUnitStatus (1.3.6.1.3.94.1.6.1.6)

Overall status of the connectivity unit. The goal of this object is to be the single poll point to check the status of the connunit. If any other component has warning, this must be set to warning. Any of these values can occur with any of the ConnUnitState values.

- Syntax—INTEGER
- Access—Read-only
- Status—Mandatory
- Return value—Refer to [Table 6](#) for connectivity unit return values. Return value is OK (3), unless one or more of the following occurs.

Table 6: Connectivity unit return values

Status	Return value
If one power supply is reporting Bad and/or not installed	warning (4)
If both power supplies are reporting Bad and/or not installed	failed (5)
If one or more cooling fan failed	warning (4)
If all cooling fans failed	failed (5)
If temperature status = "Warm"	warning (4)
If temperature status = "Overheating"	failed (5)
If any port down	warning (4)
If POST failed	failed (5)
If switch Offline or in Diagnostics mode	warning (4)

connUnitProduct (1.3.6.1.3.94.1.6.1.7)

The sml attribute Oper.Switch.Type.

- Syntax—DisplayString (SIZE (0..79))
- Access—Read-only
- Status—Mandatory
- Return value—Returns the switch product type HP StorageWorks 2/8q FC Switch

connUnitSn (1.3.6.1.3.94.1.6.1.8)

The serial number for this connectivity unit.

- Syntax—DisplayString (SIZE (0..79))
- Access—Read-only
- Status—Mandatory
- Return value—Chassis serial number

connUnitUpTime (1.3.6.1.3.94.1.6.1.9)

The number of centiseconds since the last unit initialization.

- Syntax—TimeTicks
- Access—Read-only
- Status—Mandatory
- Return value—Time interval since either POST or a reset (not including hotreset command for the NDCLA feature). POST (Power-On Self-Test) occurs during Power-On, or hardreset.

connUnitUrl (1.3.6.1.3.94.1.6.1.10)

URL to launch a management application, if applicable. Otherwise, it's an empty string. In a standalone unit, this is the same as the top-level URL. This has the same definition as systemURL for keywords. If write is not supported, the return value is invalid. This value is retained across boots.

- Syntax—DisplayString
- Access—Read-write
- Status—Mandatory
- Return value—Switch IP address (for example, http://10.0.0.1); Writes not supported, returns 'NoSuchName'.

connUnitDomainId (1.3.6.1.3.94.1.6.1.11)

24-bit Fibre Channel address ID of this connectivity unit, right justified with leading zeros if required. This must be set to the Fibre Channel address ID, or if it is a switch, it must be set to the Domain Controller address. If this value is not applicable, return all bits set to one.

- Syntax—OCTET STRING (SIZE(3))
- Access—Read-only
- Status—Mandatory
- Return value—Domain controller address (for example, FF FC 65)

connUnitProxyMaster (1.3.6.1.3.94.1.6.1.12)

A value of “yes” means this is the proxy master unit for a set of managed units. For example, this could be the only unit with a management card in it for a set of units. A standalone unit returns “yes” for this object.

- Syntax—INTEGER { unknown(1); no(2); yes(3) }
- Access—Read-only
- Status—Mandatory
- Return value—If out-of-band switch, returns yes (3); if in-band switch, returns no (2)

connUnitPrincipal (1.3.6.1.3.94.1.6.1.13)

Whether this connectivity unit is the principal unit within the group of fabric elements. If this value is not applicable, the return is unknown.

- Syntax—INTEGER { unknown(1), no(2), yes(3) }
- Access—Read-only
- Status—Mandatory
- Return value—For the principal switch, returns yes (3); otherwise returns no (2)

connUnitNumSensors (1.3.6.1.3.94.1.6.1.14)

Number of sensors in the connUnitSensorTable elements. If this value is not applicable, return unknown.

- Syntax—INTEGER
- Access—Read-only
- Status—Mandatory
- Return value—Returns the number of sensors listed in the connUnitSensorTable. For example, if there are 2 fans, 2 power supplies, and 2 temperature sensors on the switch, the return value is 6. HP StorageWorks 2/8q FC Switch = 5.

connUnitStatusChangeTime (1.3.6.1.3.94.1.6.1.15)

The sysuptime timestamp, in centiseconds, the last status change occurred.

- Syntax—TimeTicks
- Access—Read-only
- Status—Obsolete
- Return value—Obsolete; always returns error status "NoSuchName"

connUnitConfigurationChangeTime (1.3.6.1.3.94.1.6.1.16)

The sysuptime timestamp, in centiseconds, the last configuration change occurred.

- Syntax—TimeTicks
- Access—Read-only
- Status—Obsolete
- Return value—Obsolete; always returns error status "NoSuchName"

connUnitNumRevs (1.3.6.1.3.94.1.6.1.17)

The number of revisions in the connUnitRevsTable.

- Syntax—INTEGER
- Access—Read-only
- Status—Mandatory
- Return value—Number of entries in the revision table (for example, HP StorageWorks 2/8q FC Switch= 4). The revision table contains the revision numbers of all components of the switch.

connUnitNumZones (1.3.6.1.3.94.1.6.1.18)

Number of zones defined in connUnitZoneTable.

- Syntax—INTEGER
- Access—Read-only
- Status—Obsolete
- Return value—This object is obsolete; always returns error status "NoSuchName"

connUnitModuleId (1.3.6.1.3.94.1.6.1.19)

Unique ID, persistent between boots, that can be used to group a set of connUnits together into a module. Use it to create a connUnit with a connUnitType of “module” to represent a physical or logical group of connectivity units. Then, the value of the group can be set to the value of connUnitId for this “container” connUnit. connUnitModuleId must be zeros if this connUnit is not part of a module.

- Syntax—FcGlobalId
- Access—Read-only
- Status—Mandatory
- Return value—World Wide Name of the switch followed by 8 bytes of zeros. For example: 10 00 00 C0 DD 00 71 C9 00 00 00 00 00 00 00.

connUnitName (1.3.6.1.3.94.1.6.1.20)

A display string containing a name for this connectivity unit. This object value must be persistent between boots.

- Syntax—DisplayString (SIZE(0..79))
- Access—Read-write
- Status—Mandatory
- Return value—Symbolic Name of switch. The default is HP StorageWorks 2/8q FC Switch.

connUnitInfo (1.3.6.1.3.94.1.6.1.21)

A display string containing information about this connectivity unit. This object value must be persistent between boots.

- Syntax—DisplayString
- Access—Read-write
- Status—Mandatory
- Return value—Returns the ConfigDescription field for the switch. The default is HP StorageWorks 2/8q FC Switch.

connUnitControl (1.3.6.1.3.94.1.6.1.22)

This object is used to control the addressed connUnit. “Cold Start” and “Warm Start” are as defined in MIB-II and are not meant to be a factory reset.

- resetConnUnitColdStart—Addressed unit performs a “Cold Start” reset.
- resetConnUnitWarmStart—Addressed unit performs a “Warm Start” reset.
- offlineConnUnit—Addressed unit puts itself into an implementation dependant “offline” state. In general, if a unit is in an offline state, it cannot be used to perform meaningful Fibre Channel work.
- onlineConnUnit—Addressed unit puts itself into an implementation dependant “online” state. In general, if a unit is in an online state, it is capable of performing meaningful Fibre Channel work.

Each implementation can chose not to allow any or all of these values on a SET.

- Syntax—INTEGER { unknown(1); invalid(2); resetConnUnitColdStart(3); resetConnUnitWarmStart(4); offlineConnUnit(5); onlineConnUnit(6) }
- Access—Read-write
- Status—Mandatory
- Return value—Refer to the following tables for connUnitControl values.

Table 7: connUnitControl read return values

Switch setting	Return value
Online	Online (6)
Offline	Offline (5)
Diagnostics	Offline (5)
Other	Unknown (1)

Table 8: connUnitControl write control values

Control value	Result
Cold Reset (3)	Reset
Offline (5)	Offline
Online (6)	Online
Other	Not supported

connUnitContact (1.3.6.1.3.94.1.6.1.23)

Contact information for this connectivity unit, and is persistent across boots.

- Syntax—DisplayString (SIZE (0..79))
- Access—Read-write
- Status—Mandatory
- Return value—Default is <sysContact undefined>. The string size is limited to a maximum of 64.

connUnitLocation (1.3.6.1.3.94.1.6.1.24)

Location information for this connectivity unit, and is persistent across boots.

- Syntax—DisplayString (SIZE (0..79))
- Access—Read-write
- Status—Mandatory
- Return value—Default is <sysLocation undefined>. The string size is limited to a maximum of 64.

connUnitEventFilter (1.3.6.1.3.94.1.6.1.25)

This value defines the event severity that is logged by this connectivity unit. All events of severity less than or equal to connUnitEventFilter are logged in connUnitEventTable.

- Syntax—FcEventSeverity
- Access—Read-write
- Status—Mandatory
- Return value—Switch log level setting. Refer to the following tables for connUnitEventFilter values.

Table 9: connUnitEventFilter read return values

Severity levels	Return value
Critical	Critical (4)
Warn	Warning (6)
Info	Info (8)
None	Unknown (1)

Table 10: connUnitEventFilter control write values

Control value	Result
Emergency (2)	Critical
Alert (3)	Critical
Critical (4)	Critical
Error (5)	Warn
Warning (6)	Warn
Notify (7)	Info
Info (8)	Info
Debug (9)	Info
Mark (10)	Info
Unknown (1)	None

connUnitNumEvents (1.3.6.1.3.94.1.6.1.26)

Number of events currently in the connUnitEventTable.

- Syntax—INTEGER
- Access—Read-only
- Status—Mandatory
- Return value—Integer indicating the number of events in the event table

connUnitMaxEvents (1.3.6.1.3.94.1.6.1.27)

Maximum number of events that can be defined in connUnitEventTable.

- Syntax—INTEGER
- Access—Read-only
- Status—Mandatory
- Return value—Always returns 30

connUnitEventCurrID (1.3.6.1.3.94.1.6.1.28)

The last used event ID (connUnitEventIndex).

- Syntax—INTEGER
- Access—Read-only
- Status—Mandatory
- Return value—Event ID of the last event

connUnitFabricID (1.3.6.1.3.94.1.6.1.29)

A globally unique value to identify the fabric that this ConnUnit belongs to, otherwise empty string. This would typically be equal to the connUnitGlobalID of the primary switch in a Fibre Channel fabric.

- Syntax—FcGlobalId
- MaxAccess—Read-only
- Status—Mandatory
- Return value—Returns the World Wide Name of the principal switch followed by 8 bytes of zeros (for example: 10 00 00 C0 DD 00 71 C9 00 00 00 00 00 00 00)

connUnitNumLinks (1.3.6.1.3.94.1.6.1.30)

The number of links in the link table.

- Syntax—INTEGER
- MaxAccess—Read-only
- Status—Mandatory
- Return value—Returns the number of link table entries for each switch

connUnitVendorId (1.3.6.1.3.94.1.6.1.31)

The connectivity unit vendor's name.

- Syntax—DisplayString (SIZE (0..79))
- Access—Read-only
- Status—Mandatory
- Return value—HP

Revision table

The objects described in this section are in a table format indexed by World Wide Name and Index. The Table of revisions for hardware and software elements. There are four revision items in each switch. An example of how to access one of these objects given a WWN of 100000c0dd0090a7 is:

```
"snmpget localhost public
fcmgmt.connSet.connUnitRevsTable.connUnitRevsEntry.connUnitRevsUnitId.16.
0.0.192.221.0.144.167.0.0.0.0.0.0.0.0.1".
```

The number of entries in this table is variable depending on which platform is being examined and the number of blades installed. SNMP first reports the firmware revision and flasher shell version. It iterates through each of the installed blades reporting the PCB revision and ASIC version.

connUnitRevsUnitId (1.3.6.1.3.94.1.7.1.1)

The connUnitId of the connectivity unit that contains this revision table.

- Syntax—FcGlobalId
- Access—Read-only
- Status—Mandatory
- Return value—Returns the World Wide Name of the switch followed by 8 bytes of zeros (for example: 10 00 00 C0 DD 00 71 C9 00 00 00 00 00 00 00)

connUnitRevsIndex (1.3.6.1.3.94.1.7.1.2)

A unique value among all connUnitRevsEntrys with the same value of connUnitRevsUnitId, in the range between 1 and connUnitNumRevs[connUnitRevsUnitId].

- Syntax—INTEGER (1..2147483647)
- Access—Read-only
- Status—Mandatory
- Return value—Revision table index

connUnitRevsRevId (1.3.6.1.3.94.1.7.1.3)

A vendor-specific string identifying a revision of a component of the connUnit indexed by connUnitRevsUnitId.

- Syntax—DisplayString
- Access—Read-only
- Status—Mandatory
- Return value—Refer to [Table 11](#) for connUnitRevsRevId return values.

Table 11: ConnUnitRevsRevId return values

Table index	Return value
1	Active Firmware Image
2	Flasher Shell Version
3	Hardware PCB Version
4	Hardware ASIC Version

connUnitRevsDescription (1.3.6.1.3.94.1.7.1.4)

Description of a component to which the revision corresponds.

- Syntax—DisplayString
- Access—Read-only
- Status—Mandatory
- Return value—Refer to [Table 12](#) for connUnitRevsDescription return values.

Table 12: ConnUnitRevsDescription return values

Table index	Return value
1	Active Firmware Version
2	Flasher Shell Version
3	Hardware PCB Version
4	Hardware ASIC Version

Sensor table

The objects described in this section are in a table format indexed by World Wide Name and Index. The Index is the sensor number being interrogated. There are six sensor items in each switch. An example of how to access one of these objects given a WWN of 100000c0dd0090a7 is:

```
"snmpget localhost public
fcmgmt.connSet.connUnitSensorTable.connUnitSensorEntry.connUnitSensorUnit
Id.16.0.0.192.221.0.144.167.0.0.0.0.0.0.0.1".
```

connUnitSensorUnitId (1.3.6.1.3.94.1.8.1.1)

The connUnitId of the connectivity unit that contains this sensor table.

- Syntax—FcGlobalId
- Access—Read-only
- Status—Mandatory
- Return value—Returns the World Wide Name of the switch followed by 8 bytes of zeros (for example: 10 00 00 C0 DD 00 71 C9 00 00 00 00 00 00 00 00)

connUnitSensorIndex (1.3.6.1.3.94.1.8.1.2)

A unique value among all connUnitSensorEntrys with the same value of connUnitSensorUnitId, in the range between 1 and connUnitNumSensor[connUnitSensorUnitId].

- Syntax—INTEGER (1..2147483647)
- Access—Read-only
- Status—Mandatory
- Return value—Sensor table index

connUnitSensorName (1.3.6.1.3.94.1.8.1.3)

A textual identification of the sensor intended primarily for operator use.

- Syntax—DisplayString
- Access—Read-only
- Status—Mandatory
- Return value—Refer to [Table 13](#) for connUnitSensorName return values.

Table 13: ConnUnitSensorName return values

Table index	Return value
1	Power Supply 1 Status
2	Fan 1 Status
3	Temperature Status
4	Temperature Sensor 1 Value
5	Temperature Sensor 2 Value

connUnitSensorStatus (1.3.6.1.3.94.1.8.1.4)

The status indicated by the sensor.

- Syntax—INTEGER { unknown(1); other(2) - the sensor indicates other than ok (warning or failure); ok(3) - the sensor indicates OK; warning(4) - the sensor indicates a warning; failed(5) - the sensor indicates failure }
- Access—Read-only
- Status—Mandatory
- Return value—Refer to the following tables for connUnitSensorStatus return values.

Table 14: ConnUnitSensorStatus return values for board temperature

Switch value	Return value
Normal	OK (3)
Warm	Warning (4)
Overheating	Failed (5)
Other	Unknown (1)

Table 15: ConnUnitSensorStatus return values for fan status

Switch value	Return value
Good	OK (3)
Bad	Failed (5)
Other	Unknown (1)

Table 16: ConnUnitSensorStatus return values for voltage status

Switch value	Return value
Good	OK (3)
Bad	Failed (5)
Other	Unknown (1)

connUnitSensorInfo (1.3.6.1.3.94.1.8.1.5)

Miscellaneous static information about the sensor, such as its serial number.

- Syntax—DisplayString
- Access—Read-only
- Status—Mandatory
- Return value—Always returns an empty string

connUnitSensorMessage (1.3.6.1.3.94.1.8.1.6)

This describes the status of the sensor as a message. It can also provide more resolution on the sensor indication. For example, “Cover temperature 1503K, above nominal operating range” ::= { connUnitSensorEntry 6 }.

- Syntax—DisplayString
- Access—Read-only
- Status—Mandatory
- Return value—Refer to [Table 17](#) for connUnitSensorMessage values.

Table 17: ConnUnitSensorMessage values

Sensor	Value
Power Supply	Good/Bad/NotInstalled
Fan	Good/Bad/NotInstalled
Temperature Status	Normal/Warm/Overheating/NotInstalled
Temperature Value	Degrees in C

connUnitSensorType (1.3.6.1.3.94.1.8.1.7)

The type of component being monitored by this sensor.

- Syntax—INTEGER { unknown(1); other(2); battery(3); fan(4); power-supply(5); transmitter(6); enclosure(7); board(8); receiver(9) }
- Access—Read-only
- Status—Mandatory
- Return value—Refer to [Table 18](#) for connUnitSensorType return values.

Table 18: ConnUnitSensorType return values

Sensor	Value
Temperature	Board (8)
Fan	Fan (4)
Power Supply	Power Supply (5)
Voltage	Board (8)

connUnitSensorCharacteristic (1.3.6.1.3.94.1.8.1.8)

The characteristics being monitored by this sensor.

- Syntax—INTEGER { unknown(1); other(2); temperature(3); pressure(4); emf(5); currentValue(6) - current is a keyword; airflow(7); frequency(8); power(9); door(10) }
- Access—Read-only
- Status—Mandatory
- Return value—Refer to [Table 19](#) for connUnitSensorCharacteristic values.

Table 19: ConnUnitSensorCharacteristic values

Sensor	Value
Temperature Value	Temperature (3)
Temperature Status	Temperature (3)
Fan	Airflow (7)
Power Supply	Power (9)

Port table

The objects described in this section are in a table format indexed by World Wide Name and Index. The Index is the port number being interrogated. There can be different numbers of ports in each switch so the agent must determine the maximum allowable index on a switch by switch basis. An example of how to access one of these objects given a WWN of 100000c0dd0090a7 is:

```
"snmpget localhost public
fcmgmt.connSet.connUnitPortTable.connUnitPortEntry.connUnitPortUnitId.16.
0.0.192.221.0.144.167.0.0.0.0.0.0.0.1".
```

connUnitPortUnitId (1.3.6.1.3.94.1.10.1.1)

The connUnitId of the connectivity unit that contains this port.

- Syntax—FcGlobalId
- Access—Read-only
- Status—Mandatory
- Return value—Returns the World Wide Name of the switch followed by 8 bytes of zeros (for example: 10 00 00 C0 DD 00 71 C9 00 00 00 00 00 00 00).

connUnitPortIndex (1.3.6.1.3.94.1.10.1.2)

A unique value among all connUnitPortEntrys on this connectivity unit, between 1 and connUnitNumPort[connUnitPortUnitId].

- Syntax—INTEGER (1..2147483647)
- Access—Read-only
- Status—Mandatory
- Return value—Index for each port on the switch. HP StorageWorks 2/8q FC Switch= 1 - 8.

connUnitPortType (1.3.6.1.3.94.1.10.1.3)

The port type.

- Syntax—INTEGER { unknown(1); other(2); not-present(3); hub-port(4); n-port(5) - end port for fabric; nl-port(6) - end port for loop; fl-port(7) - public loop; f-port(8) - fabric port; e-port(9) - fabric expansion port; g-port(10) - generic fabric port; domain-ctl(11) - domain controller; hub-controller(12); scsi(13) - parallel SCSI port; escon(14); lan(15); wan(16); ac(17) - AC power line; dc(18) - DC power line; ssa(19) - serial storage architecture; wdm(20) - optical wave division multiplex; ib (21) - Infiniband; ipstore(22) - IP storage }
- Access—Read-only
- Status—Mandatory
- Return value—Refer to [Table 20](#) for connUnitPortType return values.

Table 20: ConnUnitPortType return values

Switch port type	Return value
G	g-port (10)
FL	fl-port (7)
F	f-port (8)
E	e-port (9)
Donor	other (2)
Other	unknown (1)

connUnitPortFCClassCap (1.3.6.1.3.94.1.10.1.4)

Bit mask that specifies the classes of service capability of this port. If this is not applicable, return all bits set to zero.

The bits have the following definition:

- unknown—0
- class-f—1
- class-one—2
- class-two—4
- class-three—8
- class-four—16
- class-five—32
- class-six—64
- Syntax—OCTET STRING (SIZE (2))
- Access—Read-only
- Status—Mandatory
- Return value—Always returns 0x0d (Class f, Class 2, and Class 3)

connUnitPortFCClassOp (1.3.6.1.3.94.1.10.1.5)

Bit mask that specifies the classes of service that are currently operational. If this is not applicable, return all bits set to zero. This object has the same definition as connUnitPortFCClassCap" ::= { connUnitPortEntry 5 }.

- Syntax—OCTET STRING (SIZE (2))
- Access—Read-only
- Status—Mandatory
- Return value—If F or FL, returns 0x0c (Class 2, and Class 3); otherwise returns 0x0d (Class f, Class 2, and Class 3)

connUnitPortState (1.3.6.1.3.94.1.10.1.5)

The user selected state of the port hardware.

- Syntax—INTEGER { unknown(1); online(2) - available for meaningful work; offline(3) - not available for meaningful work; bypassed(4) - no longer used (4/12/00); diagnostics(5) }
- Access—Read-only
- Status—Mandatory
- Return value—Refer to [Table 21](#) for connUnitPortState return values.

Table 21: ConnUnitPortState return values

Port value	Return value
Online	online (2)
Offline	offline (3)
Downed	offline (3)
Diagnostics	diagnostics (5)
Other	unknown (1)

connUnitPortStatus (1.3.6.1.3.94.1.10.1.7)

An overall protocol status for the port. This value of connUnitPortState is not online, this is reported unknown.

- Syntax—INTEGER { unknown(1); unused(2) - device cannot report this status; ready(3) - FCAL Loop or FCPH Link reset protocol; initialization complete warning(4) - do not use (4/12/00); failure(5) - do not use (4/12/00); notparticipating(6) - loop not participating and does not have a loop address; initializing(7) - protocol is proceeding; bypass(8) - do not use (4/12/00); ols(9) - FCP offline status; other(10) - status not described above }
- Access—Read-only
- Status—Mandatory
- Return value—Always returns unused (2)

connUnitPortTransmitterType (1.3.6.1.3.94.1.10.1.8)

The technology of the port transceiver.

- Syntax—INTEGER { unknown(1); other(2); unused(3); shortwave(4); longwave(5); copper(6); scsi(7); longwaveNoOFC(8); shortwaveNoOFC(9); longwaveLED(10); ssa(11) }
- Access—Read-only
- Status—Mandatory
- Return value—Refer to [Table 22](#) for connUnitPortTransmitterType return values.

Table 22: ConnUnitPortTransmitterType return values

SFP transmitter type	Return value
Not Installed	Unused (3)
SL	Shortwave (4)
LL	Longwave (5)
LC	LongwaveNoOFC (8)
SN	ShortwaveNoOFC (9)
EL	Copper (6)
Other	Unknown (1)

connUnitPortModuleType (1.3.6.1.3.94.1.10.1.9)

The module type of the port connector.

- Syntax—INTEGER { unknown(1); other(2); gbic(3); embedded(4) - fixed (oneXnine); glm(5); gbicSerialId(6); gbicNoSerialId(7); gbicNotInstalled(8); smallFormFactor(9) - this is generically a small form factor connector }
- Access—Read-only
- Status—Mandatory
- Return value—Refer to [Table 23](#) for connUnitPortModuleType return values.

Table 23: ConnUnitPortModuleType return values

Type	Value
1 Gb/2Gb Ports	smallFormFactor(9)
10 Gb Ports	Other (2)

connUnitPortWwn (1.3.6.1.3.94.1.10.1.10)

The World Wide Name of the port, if applicable, otherwise returns all zeros.

- Syntax—FcGlobalId
- Access—Read-only
- Status—Mandatory
- Return value—Returns the Port World Wide Name followed by 8 bytes of zeros (for example, the return value for port #2 would be 20 02 00 C0 DD 00 71 C9 00 00 00 00 00 00 00, and the return value for port #2 would be 20 0E 00 C0 DD 00 71 C9 00 00 00 00 00 00 00). If a port is configured as a Donor, return value = 0.

connUnitPortFCId (1.3.6.1.3.94.1.10.1.11)

This is the assigned Fibre Channel ID of this port. This value is expected to be a Big Endian value of 24 bits. If this is a loop, it is the ALPA that is connected. If this is an E-Port, it only contains the domain ID left justified, zero filled. If this port does not have a Fibre Channel address, returns all bits set to 1.

- Syntax—FcAddressId
- Access—Read-only
- Status—Mandatory
- Return value—Address for each port based on Domain, Area, and ALPA (for example, port #15 would be equal to 640F00 (Domain = 0x64, Area = 0x0F, ALPA = 0x00))

connUnitPortSn (1.3.6.1.3.94.1.10.1.12)

The serial number of the unit. If not applicable, returns an empty string.

- Syntax—DisplayString (SIZE(0..79))
- Access—Read-only
- Status—Unsupported
- Return value—Always returns error status "NoSuchName"

connUnitPortRevision (1.3.6.1.3.94.1.10.1.13)

The port revision. For example, for a GBIC.

- Syntax—DisplayString (SIZE(0..79))
- Access—Read-only
- Status—Unsupported
- Return value—Always returns error status "NoSuchName"

connUnitPortVendor (1.3.6.1.3.94.1.10.1.14)

The port vendor. For example, for a GBIC.

- Syntax—DisplayString (SIZE(0..79))
- Access—Read-only
- Status—Unsupported
- Return value—Always returns error status "NoSuchName"

connUnitPortSpeed (1.3.6.1.3.94.1.10.1.15)

The speed of the port in kilobytes per second.

- Syntax—INTEGER
- Access—Read-only
- Status—Mandatory
- Return value—Operational speed, otherwise returns the administrative speed setting. If 1 Gbps, returns 106250. If 2 Gbps, returns 212500. If 10 Gbps, returns 1062500.

connUnitPortControl (1.3.6.1.3.94.1.10.1.16)

This object is used to control the addressed connUnit's port.

- **resetConnUnitPort**—If the addressed connUnit allows this operation to be performed to this port, the addressed port performs a vendor-specific "reset" operation. Examples of these operations are: the Link Reset protocol, the Loop Initialization protocol, or a re-synchronization occurring between the transceiver in the addressed port to the transceiver that the port is connected to.
- **bypassConnUnitPort**—If the addressed connUnit allows this operation to be performed to this port, the addressed port performs a vendor-specific "bypass" operation. Examples of these operations are transitioning from online to offline, a request (non-participating) command to the Loop Port state machine, or removal of the port from an arbitrated loop by a hub.
- **unbypassConnUnitPort**—If the addressed connUnit allows this operation to be performed to this port, the addressed port performs a vendor-specific "unbypass" operation. Examples of these operations are the Link Failure protocol, a request (participating) command to the Loop Port state machine, or addition of the port to an arbitrated loop by a hub.
- **offlineConnUnitPort**—If the addressed connUnit allows this operation to be performed to this port, the addressed port performs a vendor-specific "offline" operation. Examples of these operations are disabling a port's transceiver, the Link Failure protocol, request (non-participating) command to the Loop Port state machine, or removal of the port from an arbitrated loop by a hub.
- **onlineConnUnitPort**—If the addressed connUnit allows this operation to be performed to this port, the addressed port performs a vendor-specific "online" operation. Examples of these operations are enabling a port's transceiver, the Link Failure protocol, request (participating) command to the Loop Port state machine, or addition of the port from an arbitrated loop by a hub.
- **resetConnUnitPortCounters**—If the addressed connUnit allows this operation to be performed to this port, the addressed port statistics table counters is set to zero.

Each implementation can choose not to allow any or all of these values on a SET. On a read, if you do not support write, return invalid. Otherwise, return the last control operation attempted.

- Syntax—INTEGER { unknown(1); invalid(2); resetConnUnitPort(3); bypassConnUnitPort(4); unbypassConnUnitPort(5); offlineConnUnitPort(6); onlineConnUnitPort(7); resetConnUnitPortCounters(8) }
- Access—Read-only
- Status—Mandatory
- Return value—Refer to [Table 24](#) for connUnitPortControl read return values. Refer to [Table 25](#) for connUnitPortControl write command values.

Table 24: ConnUnitPortControl read return values

Port value	Return value
Online	online (7)
Offline	offline (6)
Diagnostics	offline (6)
Other	unknown (1)

Table 25: ConnUnitPortControl write command values

Control value	Command sent
Online (3)	online
Offline (6)	offline
ResetCounters (8)	clear counters
other	error returned

connUnitPortName (1.3.6.1.3.94.1.10.1.17)

A user-defined name for this port. If the port name is greater than 32 characters, the name will be truncated in the connunit.

- Syntax—DisplayString (SIZE (0..32))
- Access—Read-write
- Status—Mandatory
- Return value—Symbolic port name. A 1G or 2G only capable port, returns port followed by the port number. 10G ports returns 10G followed by the port number (for example, a 1G/2G port#2 returns 'Port2' and a 10G port#18 returns '10G-18' by default).

connUnitPortPhysicalNumber (1.3.6.1.3.94.1.10.1.18)

This is the internal port number this port is known by. In many implementations, this must be the same as connUnitPortIndex. Some implementations can have an internal port representation not compatible with the rules for table indexes. In that case, provide the internal representation of this port in this object. This value can also be used in the connUnitLinkPortNumberX or connUnitLinkPortNumberY objects of the connUnitLinkTable.

- Syntax—INTEGER
- Access—Read-only
- Status—Mandatory
- Return value—Physical port number

connUnitPortStatObject (1.3.6.1.3.94.1.10.1.19)

This contains the OID of the first object of the table that contains the statistics for this particular port. If this has a value of zero, there are no statistics available for this port. The port type information helps to identify the statistics objects that are found in the table.

- Syntax—OBJECT IDENTIFIER
- Access—Read-write
- Status—Deprecated
- Return value—Port object ID (1.2.6.1.3.94.4.5.1.1)

connUnitPortProtocolCap (1.3.6.1.3.94.1.10.1.20)

Bit mask that specifies the driver level protocol capability of this port. If this is not applicable, returns all bits set to zero.

The bits have the following definitions:

- unknown—0
- Loop—1
- Fabric—2
- SCSI—4
- TCP/IP—8
- VI—16
- FICON—32
 - Syntax—OCTET STRING (SIZE (2))
 - Access—Read-only
 - Status—Mandatory
 - Return value—Always returns 0x03 (Loop, Fabric)

connUnitPortProtocolOp (1.3.6.1.3.94.1.10.1.21)

Bit mask that specifies the driver level protocol(s) that are currently operational. If not applicable, return all bits set to zero. This object has the same definition as connUnitPortProtocolCap.

- Syntax—OCTET STRING (SIZE (2))
- Access—Read-only
- Status—Unsupported
- Return value—Always returns error status "NoSuchName"

connUnitPortNodeWwn (1.3.6.1.3.94.1.10.1.22)

The Node World Wide Name of the port if applicable, otherwise all zeros. This has the same value for a group of related ports. The container is defined as the largest physical entity (for example, all ports on HBAs on a host have the same node WWN. All ports on the same storage subsystem have the same node WWN." ::= { connUnitPortEntry 22 } .)

- Syntax—FcNameId
- Access—Read-only
- Status—Mandatory
- Return value—Returns the World Wide Name of the switch followed by 8 bytes of zeros (for example: 10 00 00 C0 DD 00 71 C9 00 00 00 00 00 00 00 00).

connUnitPortHWState (1.3.6.1.3.94.1.10.1.23)

The hardware detected state of the port.

- Syntax—INTEGER { unknown(1); failed(2) - port failed diagnostics; bypassed(3) - FCAL bypass, loop only; active(4) - connected to a device; loopback(5) - Port in external loopback; txfault(6) - Transmitter fault; noMedia(7) - media not installed; linkDown(8) - waiting for activity (rx sync) }
- Access—Read-only
- Status—Mandatory
- Return value—Refer to [Table 26](#) for connUnitPortHWState port state return values.

Table 26: ConnUnitPortHWState port state return values

Port state	Return value
If DiagStatus = Failed	Failed (2)
If SFP = Not Installed	NoMedia (7)
If SyncStatus = SyncAcquired	Active (4)
If SyncStatus = SyncLost	LinkDown (8)
Other	Unknown (1)

Event table

The objects described in this section are in a table format indexed by World Wide Name and Index. The maximum index is determined based on the number of events in the table. An example of how to access one of these objects given a WWN of 100000c0dd0090a7 is:

```
"snmpget localhost public
fcmgmt.connSet.connUnitEventTable.connUnitEventEntry.connUnitEventUnitId.
16.0.0.192.221.0.144.167.0.0.0.0.0.0.0.0.1".
```

connUnitEventUnitId (1.3.6.1.3.94.1.11.1.1)

The connUnitId of the connectivity unit that contains this event table.

- Syntax—FcGlobalId
- Access—Read-only
- Status—Mandatory
- Return value—World Wide Name of the switch (for example, 10 00 00 C0 DD 00 71 C9 00 00 00 00 00 00 00 00)

connUnitEventIndex (1.3.6.1.3.94.1.11.1.2)

Each connectivity unit has its own event buffer. As it wraps, it can write over previous events. This object is an index into the buffer. HP recommends that this table be read using getNexts to retrieve the initial table. The management application reads the event table at periodic intervals and determines if any new entries were added by comparing the last known index value with the current highest index value. The management application updates its copy of the event table. If the read interval is too long, it is possible that there can be events that may not be contained in the agent's internal event buffer. For example, an agent may read events 50-75. At the next read interval, connUnitEventCurrID is 189. If the management application tries to read event index 76, and the agent's internal buffer is 100 entries max, event index 76 is no longer available.

The index value is an incrementing integer starting from one every time there is a table reset. On table reset, all contents are emptied and all indexes are set to zero. When an event is added to the table, the event is assigned the next higher integer value than the last item entered into the table. If the index value reaches its maximum value, the next item entered causes the index value to roll over and start at one again.

- Syntax—INTEGER (1..2147483647)
- Access—Read-only
- Status—Mandatory
- Return value—Table index

connUnitEventId (1.3.6.1.3.94.1.11.1.3)

The internal event ID. Incremented for each event, ranging between 1 and connUnitMaxEvents. Not used as table index to simplify the agent implementation. When this reaches the end of the range specified by connUnitMaxEvents, the ID rolls over to start at one. This value is set back to one at reset. The relationship of this value to the index is that internal event ID can represent a smaller number than a 32 bit integer (for example, maximum 100 entries) and would only have a value range up to connUnitMaxEvents.

- Syntax—INTEGER
- Access—Read-only
- Status—Deprecated
- Return value—Unsupported; always returns error status "NoSuchName"

connUnitEventTime (1.3.6.1.3.94.1.11.1.4)

The real time when the event occurred. It has the following format:

DDMMYYYY HHMMSS

- DD=day number
- MM=month number
- YYYY=year number
- HH=hour number
- MM=minute number
- SS=seconds number

If not applicable, return either a NULL string or "00000000 000000".

- Syntax—DisplayString (SIZE (0..15))
- Access—Read-only
- Status—Mandatory
- Return value—Timestamp of the event

connUnitSEventTime (1.3.6.1.3.94.1.11.1.5)

This is the sysuptime timestamp when the event occurred.

- Syntax—connUnitSEventTime
- Access—Read-only
- Status—Mandatory
- Return value—Always returns error status "NoSuchName"

connUnitEventSeverity (1.3.6.1.3.94.1.11.1.6)

The event severity level.

- Syntax—FcEventSeverity
- Access—Read-only
- Status—Mandatory
- Return value—Always returns error status "NoSuchName"

connUnitEventType (1.3.6.1.3.94.1.11.1.7)

The type of this event.

- Syntax—INTEGER { unknown(1); other(2); status(3); configuration(4); topology(5) }
- Access—Read-only
- Status—Mandatory
- Return value—Always returns 3 (Status)

connUnitEventObject (1.3.6.1.3.94.1.11.1.8)

This is used with the connUnitEventType to identify the object to which the event refers. Examples include connUnitPortStatus.connUnitId.connUnitPortIndex and connUnitStatus.connUnitId.

- Syntax—OBJECT IDENTIFIER
- Access—Read-only
- Status—Mandatory
- Return value—Always returns error status "NoSuchName"

connUnitEventDescr (1.3.6.1.3.94.1.11.1.9)

The description of the event.

- Syntax—DisplayString
- Access—Read-only
- Status—Mandatory
- Return value—Event description in the form:
"[Id][timestamp][severity][module][Description]"

Link table

The objects described in this section are in a table format indexed by World Wide Name and Index. The index is an index into the link table for the switch. There can be as many link entries as there are ports. An example of how to access one of these objects given a WWN of 100000c0dd0090a7 is:

```
"snmpget localhost public
fcmgmt.connSet.connUnitLinkTable.connUnitLinkEntry.connUnitLinkId.16.
0.0.192.221.0.144.167.0.0.0.0.0.0.0.0.1".
```

If the agent is able to discover links which do not directly attach to members of its agency and its discovery algorithm gives some assurance the links are recently valid, it can include these links. Link information entered by administrative action can be included even if not validated directly if the link has at least one endpoint in this agency, but must not be included otherwise.

A connectivity unit fills the table. One method to fill is to use the RNID ELS (ANSI document 99-422v0). This allows you to query a port for the information needed for the link table.

This table is accessed either directly if the management software has an index value or via GetNexts. The values of the indexes are not required to be contiguous. Each entry created in this table is assigned an index. This relationship remains persistent until the entry is removed from the table or the system is reset. The total number of entries are defined by the size of the table.

connUnitLinkId (1.3.6.1.3.94.1.12.1.1)

The connUnitId of the connectivity unit that contains this link table.

- Syntax—connUnitLinkId
- Access—Read-only
- Status—Mandatory
- Return value—World Wide Name of the switch (for example, 10 00 00 C0 DD 00 71 C9 00 00 00 00 00 00 00 00)

connUnitLinkIndex (1.3.6.1.3.94.1.12.1.2)

This index is used to create a unique value for each entry in the link table with the same connUnitLinkId. The value can only be reused if it is not currently in use and the value is the next candidate to be used. This value wraps at the highest value represented by the size of INTEGER. This value is reset to zero when the system is reset, and the first value to be used is one.

- Syntax—INTEGER (1..2147483647)
- Access—Read-only
- Status—Mandatory
- Return value—Table index

connUnitLinkNodeIdX (1.3.6.1.3.94.1.12.1.3)

The node WWN of the unit at one end of the link. If the node WWN is unknown and the node is a connUnit in the responding agent, the value of this object must be equal to its connUnitID.

- Syntax—OCTET STRING (SIZE(16))
- Access—Read-only
- Status—Mandatory
- Return value—World Wide Name of the local switch for each entry in the link table (for example, 10 00 00 C0 DD 00 71 C9 00 00 00 00 00 00 00 00)

connUnitLinkPortNumberX (1.3.6.1.3.94.1.12.1.4)

The port number on the unit specified by connUnitLinkNodeIdX if known; otherwise -1. If the value is non-negative, it is equal to connUnitPortPhysicalNumber.

- Syntax—INTEGER
- Access—Read-only
- Status—Mandatory
- Return value—Local port number for each entry in the link table

connUnitLinkPortWwnX (1.3.6.1.3.94.1.12.1.5)

The port WWN of the unit specified by connUnitLinkNodeIdX if known; otherwise 16 octets of binary 0" ::= { connUnitLinkEntry 5 }.

- Syntax—connUnitLinkPortWwnX
- Access—Read-only
- Status—Mandatory
- Return value—Local World Wide port number for each entry in the link table

connUnitLinkNodeIdY (1.3.6.1.3.94.1.12.1.6)

The node WWN of the unit at the other end of the link. If the node WWN is unknown and the node is a connUnit in the responding SNMP agency, the value of this object must be equal to its connUnitID.

- Syntax—OCTET STRING (SIZE(16))
- Access—Read-only
- Status—Mandatory
- Return value—Remote World Wide Node number for each entry in the link table

connUnitLinkPortNumberY (1.3.6.1.3.94.1.12.1.7)

The port number on the unit specified by connUnitLinkNodeIdY if known; otherwise -1. If the value is non-negative, it is equal to connUnitPortPhysicalNumber.

- Syntax—OCTET STRING (SIZE(16))
- Access—Read-only
- Status—Mandatory
- Return value—Remote port number for inter-switch link, if known; otherwise, -1 (0xFFFFFFFF)

connUnitLinkPortWwnY (1.3.6.1.3.94.1.12.1.8)

The port WWN on the unit specified by connUnitLinkNodeIdY if known; otherwise 16 octets of binary 0" ::= { connUnitLinkEntry 8 }.

- Syntax—FcGlobalId
- Access—Read-only
- Status—Mandatory
- Return value—Remote Port World Wide Name for each entry in the link table, if known

connUnitLinkAgentAddressY (1.3.6.1.3.94.1.12.1.9)

The address of an FCMGMT MIB agent for the node identified by connUnitLinkNodeIdY, if known; otherwise 16 octets of binary 0" ::= { connUnitLinkEntry 9 }.

- Syntax—OCTET STRING (SIZE(16))
- Access—Read-only
- Status—Mandatory
- Return value—Remote IP address of the remote switch, if known; otherwise, returns sixteen zeroes

connUnitLinkAgentAddressTypeY (1.3.6.1.3.94.1.12.1.10)

If connUnitLinkAgentAddressY is nonzero, it is a protocol address. ConnUnitLinkAgentAddressTypeY is the “address family number” assigned by Internet Assigned Numbers Authority (IANA) to identify the address format.

- Syntax—INTEGER
- Access—Read-only
- Status—Mandatory
- Return value—Always returns 1 (Ipv4)

connUnitLinkAgentPortY (1.3.6.1.3.94.1.12.1.11)

The IP port number for the agent. This is provided in case the agent is at a non-standard SNMP port.

- Syntax—INTEGER
- Access—Read-only
- Status—Mandatory
- Return value—Returns value of 0

connUnitLinkUnitTypeY (1.3.6.1.3.94.1.12.1.12)

Type of the Fibre Channel connectivity unit as defined in connUnitType.

- Syntax—FcUnitType
- Access—Read-only
- Status—Mandatory
- Return value—Type of remote device in the link table (for example, switch (4))

connUnitLinkConnIdY (1.3.6.1.3.94.1.12.1.13)

This is the Fibre Channel ID of this port. If the connectivity unit is a switch, this is expected to be a Big Endian value of 24 bits. If this is an E_Port, it only contains the domain ID. If this is loop, it is the ALPA that is connected. If not any of those, unknown or cascaded loop, returns all bits set to 1.

- Syntax—OCTET STRING (SIZE(3))
- Access—Read-only
- Status—Mandatory
- Return value—Remote Fibre Channel address of each entry in the link table

connUnitLinkCurrIndex (1.3.6.1.3.94.1.12.1.14)

The last used link index.

- Syntax—INTEGER
- Access—Read-only
- Status—Mandatory
- Return value—Last used link table index number

Zone table

The objects described in this section are in a table format indexed Zone number and Index. The zones are numbered 1 to connUnitZoneSetNumZones, the index represents the members within the zones.

An example of how to access one of these objects:

```
fcmgmt.connSet.connUnitZoneTable.connUnitZoneEntry.connUnitZoneIndex.1.1
```

connUnitZoneIndex (1.3.6.1.3.94.1.13.1.1)

Unique table index for each zone. Valid values are between 1 and connUnitZoneSetNumZones.

- Syntax—INTEGER (1..2147483647)
- Access—Read-only
- Status—Mandatory
- Return value—Returns index number for each zone within the active zoneset

connUnitZoneMemberIndex (1.3.6.1.3.94.1.13.1.2)

Unique table index for each zone member. Valid values are between 1 and connUnitZoneNumMembers.

- Syntax—INTEGER (1..2147483647)
- Access—Read-only
- Status—Mandatory
- Return value—Returns index number for each member within a zone

connUnitZoneSetName (1.3.6.1.3.94.1.13.1.3)

Name of the active zone set to which the zone and zone member belong.

- Syntax—DisplayString (SIZE (0..79))
- Access—Read-only
- Status—Mandatory
- Return value—Returns the zone set name

connUnitZoneSetNumZones (1.3.6.1.3.94.1.13.1.4)

The number of zones in the active zone set.

- Syntax—INTEGER
- Access—Read-only
- Status—Mandatory
- Return value—Returns the number of zones within the active zoneset

connUnitZoneName (1.3.6.1.3.94.1.13.1.5)

Name of the zone.

- Syntax—DisplayString (SIZE (0..79))
- Access—Read-only
- Status—Mandatory
- Return value—Returns the name of the zone

connUnitZoneCapabilities (1.3.6.1.3.94.1.13.1.6)

1-byte bit mask that specifies the zoning capabilities supported by the fabric.

- Bit 7—Soft zones supported
- Bit 6—Hard zones supported
- Bits 5-0—Reserved
 - Syntax—OCTET STRING (SIZE(1))
 - Access—Read-only
 - Status—Mandatory
 - Return value—Always returns 0xC0

connUnitZoneEnforcementState (1.3.6.1.3.94.1.13.1.7)

1-byte bit mask that specifies the current enforcement of the Zone Set.

- Bit 7—Soft zone set enforced
- Bit 6—Hard zone set enforced
- Bits 5-0—Reserved
 - Syntax—OCTET STRING (SIZE(1))
 - Access—Read-only
 - Status—Mandatory
 - Return value—Returns the zone type. Mapped as follows:
 - Soft.....0x80
 - Hard.....0x40

connUnitZoneAttributeBlock (1.3.6.1.3.94.1.13.1.8)

A variable length structure that contains extended zone attributes defined in the FC-GS-4 enhanced zone server. See FC-GS-4 draft standard for details and format of the structure. Support of this object is optional.

- Syntax—OCTET STRING (SIZE(80))
- Access—Read-only
- Status—Mandatory
- Return value—Not supported: always returns SNMP error “NoSuchName”

connUnitZoneNumMembers (1.3.6.1.3.94.1.13.1.9)

Number of zone members in the zone: connUnitZoneName.

- Syntax—INTEGER
- Access—Read-only
- Status—Mandatory
- Return value—Returns total number of members in a zone

connUnitZoneMemberIdType (1.3.6.1.3.94.1.13.1.10)

Type of zone member ID:

- 1—Port WWN
- 2—Domain & Port ID
- 3—FC Address
- 4—Node WWN
- 5—Alias Name
- 6—'FF'h - Vendor specified.
 - Syntax—INTEGER
 - Access—Read-only
 - Status—Mandatory
 - Return value—Retrieves the member ID type. Mapped as follows:
 - WWN.....0x01 // Port WWN
 - Domain/Port.....0x02 // Domain and Port ID
 - FCaddress.....0x03 // FC Address
 - [other].....0xff // Vendor specific

connUnitZoneMemberID (1.3.6.1.3.94.1.13.1.11)

ID of the zone member based on connUnitZoneMemberIdType.

- Syntax—FcGlobalId
- Access—Read-only
- Status—Mandatory
- Return value—Returns the zone member name as a 16 8-bit octets. Mapped as follows:
 - WWN member—WWN (8 bytes) followed by 8 bytes of zeros
 - FC address—FC address (3 bytes) followed by 13 bytes of zeros
 - Domain/Port—Domain/Port address (2 bytes) followed by 14 bytes of zeros

Zoning alias table

The objects described in this section are in a table format indexed by Alias Number and Index. The aliases are numbered 1 to connUnitZoningAliasNumAliases, the index represents the members within the alias. An example of how to access one of these objects:

```
fcmgmt.connSet.connUnitZoneTable.connUnitZoneEntry.connUnitZoningAliasIndex.1.1
```

connUnitZoningAliasIndex (1.3.6.1.3.94.1.14.1.1)

Unique table index for each alias. Valid values are between 1 and connUnitZoningAliasNumAliases.

- Syntax—INTEGER (1..2147483647)
- Access—Read-only
- Status—Mandatory
- Return value—Returns the alias index

connUnitZoningAliasMemberIndex (1.3.6.1.3.94.1.14.1.2)

Unique table index for each alias member. Valid values are between 1 and connUnitZoningAliasNumMembers.

- Syntax—INTEGER (1..2147483647)
- Access—Read-only
- Status—Mandatory
- Return value—Returns the alias member index

connUnitZoningAliasNumAliases (1.3.6.1.3.94.1.14.1.3)

The number of aliases defined in this table.

- Syntax—INTEGER
- Access—Read-only
- Status—Mandatory
- Return value—Returns number of aliases defined

connUnitZoningAliasName (1.3.6.1.3.94.1.14.1.4)

The alias name.

- Syntax—DisplayString (SIZE (0..79))
- Access—Read-only
- Status—Mandatory
- Return value—Returns Alias name

connUnitZoningAliasNumMembers (1.3.6.1.3.94.1.14.1.5)

Number of members in the alias: connUnitZoningAliasName.

- Syntax—INTEGER
- Access—Read-only
- Status—Mandatory
- Return value—Returns number of members in a defined Alias zone

connUnitZoningAliasMemberIdType (1.3.6.1.3.94.1.14.1.6)

Type of alias member ID:

- 1—Port WWN
- 2—Domain & Port ID
- 3—FC Address
- Others: Reserved
 - Syntax—INTEGER
 - Access—Read-only
 - Status—Mandatory
 - Return value—Returns the alias member ID type. Mapped as follows:
 - WWN..... 0x01 // Port WWN
 - DomainPort..... 0x02 // Domain and Port ID
 - FC Address..... 0x03 // FC Address
 - [other]..... 0xff // Vendor specific

connUnitZoningAliasMemberID (1.3.6.1.3.94.1.14.1.7)

ID of the alias member based on connUnitZoningAliasMemberIdType.

- Syntax—FcGlobalId
- Access—Read-only
- Status—Mandatory
- Return value—Returns the alias zone member name as 16 8-bit octets. Mapped as follows:
 - WWN member—WWN (8 bytes) followed by 8 bytes of zeros
 - FC address—FC address (3 bytes) followed by 13 bytes of zeros
 - Domain/Port—Domain/Port address (2 bytes) followed by 14 bytes of zeros

Port statistics table

The objects described in this section are in a table format indexed by World Wide Name and Index. The index represents the port number to interrogate. An example of how to access one of these objects given a WWN of 100000c0dd0090a7 is:

```
"snmpget localhost public
fcmgmt.statSet.connUnitPortStatTable.connUnitPortStatEntry.connUnitPortStatUnitId.16.0.0.192.221.0.144.167.0.0.0.0.0.0.0.1".
```

There is one and only one statistics table for each individual port. For all objects in statistics table, if the object is not supported by the connUnit, the high order bit is set to 1 with all other bits set to zero. The high order bit is reserved to indicate if the object is supported or not. All objects start at a value of zero at hardware initialization and continue incrementing until the end of 63 bits, and then wrap to zero.

connUnitPortStatUnitId (1.3.6.1.3.94.4.5.1.1)

A unique value among all entries in this table having the same connUnitPortStatUnitId, between 1 and connUnitNumPort [connUnitPortStatUnitId].

- Syntax—FcGlobalId
- Access—Read-only
- Status—Mandatory
- Return value—Returns the World Wide Name of the switch followed by 8 bytes of zeros (for example: 10 00 00 C0 DD 00 71 C9 00 00 00 00 00 00 00 00)

connUnitPortStatIndex (1.3.6.1.3.94.4.5.1.2)

A unique value among all entries in this table, between 0 and connUnitNumPort[connUnitPortUnitId].

- Syntax—INTEGER (0..2147483647)
- Access—Read-only
- Status—Mandatory
- Return value—Port table index

connUnitPortStatCountError (1.3.6.1.3.94.4.5.1.3)

A count of the errors that have occurred on this port.

- Syntax—OCTET STRING (SIZE (8))
- Access—Read-only
- Status—Mandatory
- Return value—Hexidecimal value indicating the total number of errors for a port

connUnitPortStatCountTxObjects (1.3.6.1.3.94.4.5.1.4)

The number of frames/packets/IOs/etc transmitted by this port. A Fibre Channel frame starts with SOF and ends with EOF. Fibre Channel loop devices must not count frames passed through. This value represents the sum total for all other Tx objects.

- Syntax—OCTET STRING (SIZE (8))
- Access—Read-only
- Status—Mandatory
- Return value—Hexidecimal value indicating the total number of bytes transmitted by a port

connUnitPortStatCountRxObjects (1.3.6.1.3.94.4.5.1.5)

The number of frames/packets/IOs/etc received by this port. A Fibre Channel frame starts with SOF and ends with EOF. Fibre Channel loop devices must not count frames passed through. This value represents the sum total for all other Rx objects.

- Syntax—OCTET STRING (SIZE (8))
- Access—Read-only
- Status—Mandatory
- Return value—Hexidecimal value indicating the total number of bytes received by a port

connUnitPortStatCountTxElements (1.3.6.1.3.94.4.5.1.6)

The number of octets or bytes that have been transmitted by this port. One second periodic polling of the port. This value is saved and compared with the next polled value to compute net throughput. For Fibre Channel, ordered sets are not included in the count.

- Syntax—OCTET STRING (SIZE (8))
- Access—Read-only
- Status—Mandatory
- Return value—Hexidecimal value indicating the total number of bytes transmitted by a port

connUnitPortStatCountRxElements (1.3.6.1.3.94.4.5.1.7)

The number of octets or bytes that have been received by this port. One second periodic polling of the port. This value is saved and compared with the next polled value to compute net throughput. For Fibre Channel, ordered sets are not included in the count.

- Syntax—OCTET STRING (SIZE (8))
- Access—Read-only
- Status—Mandatory
- Return value—Hexidecimal value indicating the total number of bytes received by a port

connUnitPortStatCountBBCreditZero (1.3.6.1.3.94.4.5.1.8)

Count of transitions in/out of BBCredit zero state. The other side is not providing any credit. This is a Fibre Channel statistic only.

- Syntax—OCTET STRING (SIZE (8))
- Access—Read-only
- Status—Mandatory
- Return value—Unsupported; always returns high order bit to 1 with all other bits set to zero

connUnitPortStatCountInputBuffersFull (1.3.6.1.3.94.4.5.1.9)

Count of occurrences when all input buffers of a port were full and outbound buffer-to-buffer credit transitioned to zero. There is no credit to provide to other side. This is a Fibre Channel statistic only.

- Syntax—OCTET STRING (SIZE (8))
- Access—Read-only
- Status—Mandatory
- Return value—Unsupported; always returns high order bit to 1 with all other bits set to zero

connUnitPortStatCountFBSYFrames (1.3.6.1.3.94.4.5.1.10)

Count of times that FBSY was returned to this port as a result of a frame that could not be delivered to the other end of the link. This occurs if either the fabric or the destination port is temporarily busy. Port can only occur on SOFc1 frames (the frames that establish a connection). This is a Fibre Channel-only statistic. This is the sum of all classes. If you cannot keep the by-class counters, keep the sum counters.

- Syntax—OCTET STRING (SIZE (8))
- Access—Read-only
- Status—Mandatory
- Return value—Hexidecimal number indicating the total number of FBusy on a port

connUnitPortStatCountPBSYFrames (1.3.6.1.3.94.4.5.1.11)

Count of times that PBSY was returned to this port as a result of a frame that could not be delivered to the other end of the link. This occurs if the destination port is temporarily busy. PBSY can only occur on SOFc1 frames (the frames that establish a connection). This is a Fibre Channel-only statistic. This is the sum of all classes. If you cannot keep the by-class counters, keep the sum counters.

- Syntax—OCTET STRING (SIZE (8))
- Access—Read-only
- Status—Mandatory
- Return value—Unsupported; always returns high order bit set to 1 with remaining bits set to zero.

connUnitPortStatCountFRJTFrames (1.3.6.1.3.94.4.5.1.12)

Count of times that FRJT was returned to this port as a result of a frame that was rejected by the fabric. This is the total for all classes and is a Fibre Channel-only statistic.

- Syntax—OCTET STRING (SIZE (8))
- Access—Read-only
- Status—Mandatory
- Return value—Hexidecimal number indicating the total number of Frame Rejects on a port

connUnitPortStatCountPRJTFrames (1.3.6.1.3.94.4.5.1.13)

Count of times that FRJT was returned to this port as a result of a frame that was rejected at the destination N_Port. This is the total for all classes and is a Fibre Channel-only statistic.

- Syntax—OCTET STRING (SIZE (8))
- Access—Read-only
- Status—Mandatory
- Return value—Unsupported; always returns high order bit to 1 with all other bits set to zero

connUnitPortStatCountClass1RxFrames (1.3.6.1.3.94.4.5.1.14)

Count of Class 1 frames received at this port. This is a Fibre Channel-only statistic.

- Syntax—OCTET STRING (SIZE (8))
- Access—Read-only
- Status—Mandatory
- Return value—Unsupported; always returns high order bit to 1 with all other bits set to zero

connUnitPortStatCountClass1TxFrames (1.3.6.1.3.94.4.5.1.15)

Count of Class 1 frames transmitted out this port. This is a Fibre Channel-only statistic.

- Syntax—OCTET STRING (SIZE (8))
- Access—Read-only
- Status—Mandatory
- Return value—Unsupported; always returns high order bit to 1 with all other bits set to zero

connUnitPortStatCountClass1FBSYFrames (1.3.6.1.3.94.4.5.1.16)

Count of times that FBSY was returned to this port as a result of a Class 1 frame that could not be delivered to the other end of the link. This occurs if either the fabric or the destination port is temporarily busy. FBSY can only occur on SOFc1 frames (the frames that establish a connection). This is a Fibre Channel-only statistic.

- Syntax—OCTET STRING (SIZE (8))
- Access—Read-only
- Status—Mandatory
- Return value—Unsupported; always returns high order bit to 1 with all other bits set to zero

connUnitPortStatCountClass1PBSYFrames (1.3.6.1.3.94.4.5.1.17)

Count of times that PBSY was returned to this port as a result of a Class 1 frame that could not be delivered to the other end of the link. This occurs if the destination N_Port is temporarily busy. PBSY can only occur on SOFc1 frames (the frames that establish a connection). This is a Fibre Channel-only statistic.

- Syntax—OCTET STRING (SIZE (8))
- Access—Read-only
- Status—Mandatory
- Return value—Unsupported; always returns high order bit to 1 with all other bits set to zero

connUnitPortStatCountClass1FRJTFrames (1.3.6.1.3.94.4.5.1.18)

Count of times that FRJT was returned to this port as a result of a Class 1 frame that was rejected by the fabric. This is a Fibre Channel-only statistic.

- Syntax—OCTET STRING (SIZE (8))
- Access—Read-only
- Status—Mandatory
- Return value—Unsupported; always returns high order bit to 1 with all other bits set to zero

connUnitPortStatCountClass1PRJTFrames (1.3.6.1.3.94.4.5.1.19)

Count of times that PRJT was returned to this port as a result of a Class 1 frame that was rejected at the destination N_Port. This is a Fibre Channel-only statistic.

- Syntax—OCTET STRING (SIZE (8))
- Access—Read-only
- Status—Mandatory
- Return value—Unsupported; always returns high order bit to 1 with all other bits set to zero

connUnitPortStatCountClass2RxFrames (1.3.6.1.3.94.4.5.1.20)

Count of Class 2 frames received at this port. This is a Fibre Channel-only statistic.

- Syntax—OCTET STRING (SIZE (8))
- Access—Read-only
- Status—Mandatory
- Return value—Total number of Class 2 frames received by a port

connUnitPortStatCountClass2TxFrames (1.3.6.1.3.94.4.5.1.21)

Count of Class 2 frames transmitted out this port. This is a Fibre Channel-only statistic.

- Syntax—OCTET STRING (SIZE (8))
- Access—Read-only
- Status—Mandatory
- Return value—Total number of Class 2 frames transmitted by a port

connUnitPortStatCountClass2FBSYFrames (1.3.6.1.3.94.4.5.1.22)

Count of times that FBSY was returned to this port as a result of a Class 2 frame that could not be delivered to the other end of the link. This occurs if either the fabric or the destination port is temporarily busy. FBSY can only occur on SOFc1 frames (the frames that establish a connection). This is a Fibre Channel-only statistic.

- Syntax—OCTET STRING (SIZE (8))
- Access—Read-only
- Status—Mandatory
- Return value—Unsupported; always returns high order bit to 1 with all other bits set to zero

connUnitPortStatCountClass2PBSYFrames (1.3.6.1.3.94.4.5.1.23)

Count of times that PBSY was returned to this port as a result of a Class 2 frame that could not be delivered to the other end of the link. This occurs if the destination N_Port is temporarily busy. PBSY can only occur on SOFc1 frames (the frames that establish a connection). This is a Fibre Channel-only statistic.

- Syntax—OCTET STRING (SIZE (8))
- Access—Read-only
- Status—Mandatory
- Return value—Unsupported; always returns high order bit to 1 with all other bits set to zero

connUnitPortStatCountClass2FRJTFrames (1.3.6.1.3.94.4.5.1.24)

Count of times that FRJT was returned to this port as a result of a Class 2 frame that was rejected by the fabric. This is a Fibre Channel-only statistic.

- Syntax—OCTET STRING (SIZE (8))
- Access—Read-only
- Status—Mandatory
- Return value—Unsupported; always returns high order bit to 1 with all other bits set to zero

connUnitPortStatCountClass2PRJTFrames (1.3.6.1.3.94.4.5.1.25)

Count of times that PRJT was returned to this port as a result of a Class 2 frame that was rejected at the destination N_Port. This is a Fibre Channel-only statistic.

- Syntax—OCTET STRING (SIZE (8))
- Access—Read-only
- Status—Mandatory
- Return value—Unsupported; always returns high order bit to 1 with all other bits set to zero

connUnitPortStatCountClass3RxFrames (1.3.6.1.3.94.4.5.1.26)

Count of Class 3 frames received at this port. This is a Fibre Channel-only statistic.

- Syntax—OCTET STRING (SIZE (8))
- Access—Read-only
- Status—Mandatory
- Return value—Total number of Class 3 frames received by a port

connUnitPortStatCountClass3TxFrames (1.3.6.1.3.94.4.5.1.27)

Count of Class 3 frames transmitted out this port. This is a Fibre Channel-only statistic.

- Syntax—OCTET STRING (SIZE (8))
- Access—Read-only
- Status—Mandatory
- Return value—Total number of Class 3 frames transmitted by a port

connUnitPortStatCountClass3Discards (1.3.6.1.3.94.4.5.1.28)

Count of Class 3 frames that were discarded upon reception at this port. There is no FBSY or FRJT generated for Class 3 frames. They are discarded if they cannot be delivered. This is a Fibre Channel-only statistic.

- Syntax—OCTET STRING (SIZE (8))
- Access—Read-only
- Status—Mandatory
- Return value—Total number of Class3Toss frames for a port

connUnitPortStatCountRxMulticastObjects (1.3.6.1.3.94.4.5.1.29)

Count of Multicast frames or packets received at this port.

- Syntax—OCTET STRING (SIZE (8))
- Access—Read-only
- Status—Mandatory
- Return value—Unsupported; always returns high order bit to 1 with all other bits set to zero

connUnitPortStatCountTxMulticastObjects (1.3.6.1.3.94.4.5.1.30)

Count of Multicast frames or packets transmitted out this port.

- Syntax—OCTET STRING (SIZE (8))
- Access—Read-only
- Status—Mandatory
- Return value—Unsupported; always returns high order bit to 1 with all other bits set to zero

connUnitPortStatCountRxBroadcastObjects (1.3.6.1.3.94.4.5.1.31)

Count of Broadcast frames or packets received at this port.

- Syntax—OCTET STRING (SIZE (8))
- Access—Read-only
- Status—Mandatory
- Return value—Unsupported; always returns high order bit to 1 with all other bits set to zero

connUnitPortStatCountTxBroadcastObjects (1.3.6.1.3.94.4.5.1.32)

Count of Broadcast frames or packets transmitted out this port. On a Fibre Channel loop, count only OPN_r frames generated.

- Syntax—OCTET STRING (SIZE (8))
- Access—Read-only
- Status—Mandatory
- Return value—Unsupported; always returns high order bit to 1 with all other bits set to zero

connUnitPortStatCountRxLinkResets (1.3.6.1.3.94.4.5.1.33)

Count of link resets. This is the number of LRs received. This is a Fibre Channel-only statistic.

- Syntax—OCTET STRING (SIZE (8))
- Access—Read-only
- Status—Mandatory
- Return value—Total number of RxLinkResets received by a port

connUnitPortStatCountTxLinkResets (1.3.6.1.3.94.4.5.1.34)

Count of link resets. The number of LRs transmitted. This is a Fibre Channel-only statistic.

- Syntax—OCTET STRING (SIZE (8))
- Access—Read-only
- Status—Mandatory
- Return value—Total number of TxLinkResets transmitted by a port

connUnitPortStatCountNumberLinkResets (1.3.6.1.3.94.4.5.1.35)

Count of link resets and LIPs detected at this port. The number times the reset link protocol is initiated. These are the count of the logical resets, and a count of the number of primitives. This is a Fibre Channel-only statistic.

- Syntax—OCTET STRING (SIZE (8))
- Access—Read-only
- Status—Mandatory
- Return value—Total number of TotalLinkResets for a port

connUnitPortStatCountRxOfflineSequences (1.3.6.1.3.94.4.5.1.36)

Count of offline primitive OLSs received at this port. This is a Fibre Channel-only statistic.

- Syntax—OCTET STRING (SIZE (8))
- Access—Read-only
- Status—Mandatory
- Return value—Total number of RxOfflineSeqs received by a port

connUnitPortStatCountTxOfflineSequences (1.3.6.1.3.94.4.5.1.37)

Count of offline primitive OLSs transmitted by this port. This is a Fibre Channel-only statistic.

- Syntax—OCTET STRING (SIZE (8))
- Access—Read-only
- Status—Mandatory
- Return value—Total number of TxOfflineSeqs transmitted by a port

connUnitPortStatCountNumberOfflineSequences (1.3.6.1.3.94.4.5.1.38)

Count of offline primitive sequences received at this port. This is a Fibre Channel-only statistic.

- Syntax—OCTET STRING (SIZE (8))
- Access—Read-only
- Status—Mandatory
- Return value—Total number of TotalOfflineSeqs received by a port

connUnitPortStatCountLinkFailures (1.3.6.1.3.94.4.5.1.39)

Count of link failures. This count is part of the Link Error Status Block (LESB). (FC-PH 29.8). This is a Fibre Channel-only statistic.

- Syntax—OCTET STRING (SIZE (8))
- Access—Read-only
- Status—Mandatory
- Return value—Total number of LinkFailures for a port

connUnitPortStatCountInvalidCRC (1.3.6.1.3.94.4.5.1.40)

Count of frames received with invalid CRC. This count is part of the Link Error Status Block (LESB). (FC-PH 29.8). Loop ports must not count CRC errors passing through when monitoring. This is a Fibre Channel-only statistic.

- Syntax—OCTET STRING (SIZE (8))
- Access—Read-only
- Status—Mandatory
- Return value—Total number of InvalidCRCs received by a port

connUnitPortStatCountInvalidTxWords (1.3.6.1.3.94.4.5.1.41)

Count of invalid transmission words received at this port. This count is part of the Link Error Status Block (LESB). (FC-PH 29.8). This is a Fibre Channel-only statistic.

- Syntax—OCTET STRING (SIZE (8))
- Access—Read-only
- Status—Mandatory
- Return value—Total number of DecodeErrors for a port.

connUnitPortStatCountPrimitiveSequenceProtocolErrors (1.3.6.1.3.94.4.5.1.42)

Count of primitive sequence protocol errors detected at this port. This count is part of the Link Error Status Block (LESB). (FC-PH 29.8). This is a Fibre Channel-only statistic.

- Syntax—OCTET STRING (SIZE (8))
- Access—Read-only
- Status—Mandatory
- Return value—Total number of PrimSeqErrors for a port

connUnitPortStatCountLossOfSignal (1.3.6.1.3.94.4.5.1.43)

Count of instances of signal loss detected at port. This count is part of the Link Error Status Block (LESB). (FC-PH 29.8). This is a Fibre Channel-only statistic.

- Syntax—OCTET STRING (SIZE (8))
- Access—Read-only
- Status—Mandatory
- Return value—Unsupported; always returns high order bit to 1 with all other bits set to zero

connUnitPortStatCountLossOfSynchronization (1.3.6.1.3.94.4.5.1.44)

Count of instances of synchronization loss detected at port. This count is part of the Link Error Status Block (LESB). (FC-PH 29.8). This is a Fibre Channel-only statistic.

- Syntax—OCTET STRING (SIZE (8))
- Access—Read-only
- Status—Mandatory
- Return value—Total number LossOfSyncs detected by this port

connUnitPortStatCountInvalidOrderedSets (1.3.6.1.3.94.4.5.1.45)

Count of invalid ordered sets received at port. This count is part of the Link Error Status Block (LESB). (FC-PH 29.8). This is a Fibre Channel-only statistic.

- Syntax—OCTET STRING (SIZE (8))
- Access—Read-only
- Status—Mandatory
- Return value—Unsupported; always returns high order bit to 1 with all other bits set to zero

connUnitPortStatCountFramesTooLong (1.3.6.1.3.94.4.5.1.46)

Count of frames received at this port where the frame length was greater than what was agreed to in FLOGI/PLOGI. This could be caused by losing the end of frame delimiter. This is a Fibre Channel-only statistic.

- Syntax—OCTET STRING (SIZE (8))
- Access—Read-only
- Status—Mandatory
- Return value—Unsupported; always returns high order bit to 1 with all other bits set to zero

connUnitPortStatCountFramesTruncated (1.3.6.1.3.94.4.5.1.47)

Count of frames received at this port where the frame length was less than the minimum indicated by the frame header (normally 24 bytes). It could be more if the DF_CTL field indicates that an optional header should have been present. This is a Fibre Channel-only statistic.

- Syntax—OCTET STRING (SIZE (8))
- Access—Read-only
- Status—Mandatory
- Return value—Unsupported; always returns high order bit to 1 with all other bits set to zero

connUnitPortStatCountAddressErrors (1.3.6.1.3.94.4.5.1.48)

Count of frames received with unknown addressing.

- Syntax—OCTET STRING (SIZE (8))
- Access—Read-only
- Status—Mandatory
- Return value—Total number of InvalidDestAddr frames received by a port

connUnitPortStatCountDelimiterErrors (1.3.6.1.3.94.4.5.1.49)

Count of invalid frame delimiters received at this port. An example is a frame with a Class 2 start and a Class 3 at the end. This is a Fibre Channel-only statistic.

- Syntax—OCTET STRING (SIZE (8))
- Access—Read-only
- Status—Mandatory
- Return value—Unsupported; always returns high order bit to 1 with all other bits set to zero

connUnitPortStatCountEncodingDisparityErrors (1.3.6.1.3.94.4.5.1.50)

Count of disparity errors received at this port. This is a Fibre Channel-only statistic.

- Syntax—OCTET STRING (SIZE (8))
- Access—Read-only
- Status—Mandatory
- Return value—Unsupported; always returns high order bit to 1 with all other bits set to zero

Simple Name Server (SNS) table

The objects described in this section are in a table format indexed by World Wide Name and Index. The index represents the table index. An example of how to access one of these objects given a WWN of 100000c0dd0090a7 is:

```
"snmpget localhost public
fcmgmt.connUnitServiceSet.connUnitServiceTables.connUnitSnsTable.connUnit
SnsEntry.connUnitSnsId.16.0.0.192.221.0.144.167.0.0.0.0.0.0.0.1".
```

The Fibre Channel Simple Name Server table contains an entry for each device presently known to this connUnit. There is not any version on this since FC-GS3 does not define a version today.

This table is accessed either directly if the management software has an index value or by using GetNexts. The value of the indexes are not required to be contiguous. Each entry created in this table is assigned an index. This relationship is retained until the entry is removed from the table or the system is reset. The total number of entries is defined by the size of the table.

connUnitSnsMaxEntry (1.3.6.1.3.94.5.1.1)

The current number of entries in the table.

- Syntax—INTEGER
- MaxAccess—Read-only
- Status—Mandatory
- Return value—Returns the number of entries registered in the Simple Name Server for all switches

connUnitSnsId (1.3.6.1.3.94.5.2.1.1.1)

The connUnitId of the connectivity unit that contains this Name Server table.

- Syntax—OCTET STRING (SIZE (16))
- Access—Read-only
- Status—Mandatory
- Return value—Returns the World Wide Name of the switch followed by 8 bytes of zeros (for example: 10 00 00 C0 DD 00 71 C9 00 00 00 00 00 00 00 00)

connUnitSnsPortIndex (1.3.6.1.3.94.5.2.1.1.2)

The physical port number of this SNS table entry. Each physical port has an SNS table with 1-n entries indexed by ConnUnitSnsPortIdentifier (port address).

- Syntax—INTEGER
- MaxAccess—Read-only
- Status—Mandatory
- Return value—Name server table index

connUnitSnsPortIdentifier (1.3.6.1.3.94.5.2.1.1.3)

The port identifier for this entry in the SNS table.

- Syntax—FcAddressId
- Access—Read-only
- Status—Mandatory
- Return value—24-bit Fibre Channel address for each entry in the name server table based on Domain, Area, and ALPA

connUnitSnsPortName (1.3.6.1.3.94.5.2.1.1.4)

The Port World Wide Name for this entry in the SNS table.

- Syntax—FcNameId
- Access—Read-only
- Status—Mandatory
- Return value—Port World Wide Name of the device in the name server table

connUnitSnsNodeName (1.3.6.1.3.94.5.2.1.1.5)

The node name for this entry in the SNS table.

- Syntax—FcNameId
- Access—Read-only
- Status—Mandatory
- Return value—Node World Wide Name of the device in the name server table

connUnitSnsClassOfSvc (1.3.6.1.3.94.5.2.1.1.6)

The classes of service offered by this entry in the SNS table.

- Syntax—OCTET STRING (SIZE (1))
- Access—Read-only
- Status—Mandatory
- Return value—Value indicating the first registered class of service for an entry in the name server table. This is a bit mask where each bit that represents the class of service is set to a value of one if the class is supported. Class 1 is bit zero.

connUnitSnsNodeIPAddress (1.3.6.1.3.94.5.2.1.1.7)

The IPv6 formatted address of the node for this entry in the SNS table.

- Syntax—OCTET STRING (SIZE (16))
- Access—Read-only
- Status—Mandatory
- Return value—Unsupported; always returns error status "NoSuchName"

connUnitSnsProcAssoc (1.3.6.1.3.94.5.2.1.1.8)

The process associator for this entry in the SNS table.

- Syntax—OCTET STRING (SIZE (16))
- Access—Read-only
- Status—Mandatory
- Return value—Unsupported; always returns error status "NoSuchName"

connUnitSnsFC4Type (1.3.6.1.3.94.5.2.1.1.9)

The FC-4 types supported by this entry in the SNS table.

- Syntax—OCTET STRING (SIZE (32))
- Access—Read-only
- Status—Mandatory
- Return value—Value indicating the FC-4 Types registered for the device in the name server table. This is a 32 byte field with each bit uniquely identifying the FC-4 Type registered as defined in FC-GS-3 specification. Example: SCSI FCP (bit 8) = 00 00 01 00.

connUnitSnsPortType (1.3.6.1.3.94.5.2.1.1.10)

The port type of this entry in the SNS table.

- Syntax—OCTET STRING (SIZE (1))
- Access—Read-only
- Status—Mandatory
- Return value—Value indicating the PortType for the entry in the name server table. Refer to [Table 27](#) for connUnitPortType port type return values.

Table 27: ConnUnitPortType state return values

Port type	Return value (hexidecimal)
N	1
NL	2
F/NL	3
NX	7F
F	8
FL	82
E	84
B	85

connUnitSnsPortIPAddress (1.3.6.1.3.94.5.2.1.1.11)

The IPv6 formatted address of this entry in the SNS table.

- Syntax—OCTET STRING (SIZE (16))
- Access—Read-only
- Status—Mandatory
- Return value—Unsupported; always returns error status "NoSuchName"

connUnitSnsFabricPortName (1.3.6.1.3.94.5.2.1.1.12)

The fabric port name of this entry in the SNS table.

- Syntax—FcNameId
- Access—Read-only
- Status—Mandatory
- Return value—Switch port Port World Wide Name for the device in the name server table

connUnitSnsHardAddress (1.3.6.1.3.94.5.2.1.1.13)

The hard ALPA of this entry in the SNS table.

- Syntax—FcAddressId
- Access—Read-only
- Status—Mandatory
- Return value—Unsupported; always returns error status "NoSuchName";

connUnitSnsSymbolicPortName (1.3.6.1.3.94.5.2.1.1.14)

The symbolic port name of this entry in the SNS table.

- Syntax—DisplayString (SIZE (0..79))
- Access—Read-only
- Status—Mandatory
- Return value—Symbolic Port Name registered by the device in the name server table. If not registered, returns (NULL).

connUnitSnsSymbolicNodeName (1.3.6.1.3.94.5.2.1.1.15)

The symbolic node name of this entry in the SNS table.

- Syntax—DisplayString (SIZE (0..79))
- Access—Read-only
- Status—Mandatory
- Return value—Symbolic node name registered by the device in the name server table. If not registered, returns (NULL).

Platform table

The Platform table is a simple, read-only view of platform registration entries. Platform registry is a service hosted by the connectivity unit, in a very similar manner as the SNS table. The platform table is contained by the connectivity unit. A platform can register its attributes and platform nodes with the registry service.

The platform table is a flat, double-indexed MIB table. To keep the table simple, only one platform management URL is exposed. If a platform registers more than one management URL, the first one is reported in this table. This table is based on the fabric configuration server defined in the FC-GS-3 standard and enhanced platform attributes proposed for FC-GS-4. Note that the information contained in this table can only contain the platforms that this connUnit can see or it can contain a fabric wide view of the platforms.

connUnitPlatformMaxEntry (1.3.6.1.3.94.5.1.2)

The maximum number of entries in the platform table.

- Syntax—INTEGER
- Access—Read-only
- Status—Mandatory
- Return value—Unsupported; always returns error status "NoSuchName"

connUnitPlatformIndex (1.3.6.1.3.94.5.2.2.1.1)

Unique table index for each platform. Valid values are between 1 and connUnitPlatformsMaxEntry.

- Syntax—INTEGER (1..2147483647)
- Access—Read-only
- Status—Mandatory
- Return value—Unsupported; always returns error status "NoSuchName"

connUnitPlatformNodeIndex (1.3.6.1.3.94.5.2.2.1.2)

Unique table index for each platform node. Valid values are between 1 and connUnitPlatformsNumNodes.

- Syntax—INTEGER (1..2147483647)
- Access—Read-only
- Status—Mandatory
- Return value—Unsupported; always returns error status "NoSuchName"

connUnitPlatformUnitID (1.3.6.1.3.94.5.2.2.1.3)

The connUnitId of the connectivity unit that contains this Platform table.

- Syntax—FcGlobalId
- Access—Read-only
- Status—Mandatory
- Return value—Unsupported; always returns error status "NoSuchName"

connUnitPlatformName (1.3.6.1.3.94.5.2.2.1.4)

The platform name. can be either a readable string or a unique ID format as specified in the FC-GS-4 draft standard.

- Syntax—OCTET STRING (SIZE(79))
- Access—Read-only
- Status—Mandatory
- Return value—Unsupported; always returns error status "NoSuchName"

connUnitPlatformType (1.3.6.1.3.94.5.2.2.1.6)

The platform type.

- Syntax—FcUnitType
- Access—Read-only
- Status—Mandatory
- Return value—Unsupported; always returns error status "NoSuchName"

connUnitPlatformLabel (1.3.6.1.3.94.5.2.2.1.7)

An administratively assigned symbolic name for the platform. The Platform label must only contain print-able ASCII characters.

- Syntax—DisplayString (SIZE (0..79))
- Access—Read-only
- Status—Mandatory
- Return value—Unsupported; always returns error status "NoSuchName"

connUnitPlatformDescription (1.3.6.1.3.94.5.2.2.1.8)

A textual description of the platform. This value must include the full name and version identification of the platform's hardware type and software operating system. The Platform description must only contain printable ASCII characters.

- Syntax—DisplayString (SIZE (0..79))
- Access—Read-only
- Status—Mandatory
- Return value—Unsupported; always returns error status "NoSuchName"

connUnitPlatformLocation (1.3.6.1.3.94.5.2.2.1.9)

The physical location of the platform (e.g., telephone closet, 3rd floor). The Platform Location shall only contain printable ASCII characters.

- Syntax—DisplayString (SIZE (0..79))
- Access—Read-only
- Status—Mandatory
- Return value—Unsupported; always returns error status "NoSuchName"

connUnitPlatformManagementUrl (1.3.6.1.3.94.5.2.2.1.10)

Primary management URL for the platform. If the platform registers more than one URL, this URL is equal to the first in the list.

- Syntax—DisplayString (SIZE (0..79))
- Access—Read-only
- Status—Mandatory
- Return value—Unsupported; always returns error status "NoSuchName"

connUnitPlatformNumNodes (1.3.6.1.3.94.5.2.2.1.11)

Number of nodes contained in the platform.

- Syntax—INTEGER
- Access—Read-only
- Status—Mandatory
- Return value—Unsupported; always returns error status "NoSuchName"

connUnitPlatformNodeName (1.3.6.1.3.94.5.2.2.1.12)

The name (WWN - world wide name) of the node contained by the platform.

- Syntax—FcGlobalId
- Access—Read-only
- Status—Read-only.
- Return value—Unsupported; always returns error status "NoSuchName"

Trap table

Traps are asynchronous messages sent from the agent (residing on the switch) to the manager (residing on the workstation) to identify significant events.

There can be up to 5 trap addresses within the trap table. All trap information is stored within the switch and is accessible to Telnet and the SNMP agent, and is persistent between boots. An example of how to access one of these objects given an IP address of 10.32.165.4 is:

```
"snmpget localhost public
fcmgmt.trapReg.trapRegTable.trapRegEntry.trapRegFilter.10.32.165.4.162".
```

A trap event is reported when the incoming error has a severity level less than or equal to the configured severity level. The trap event types and trap severity levels are listed in [Table 28](#).

Table 28: Trap severity levels

Event type	Severity level
Unknown	1
Emergency	2
Alert	3
Critical	4
Error	5
Warning	6
Notify	7
Info	8
Debug	9
Mark	10

trapMaxClients (1.3.6.1.3.94.2.1)

The maximum number of SNMP trap recipients supported by the connectivity unit.

- Syntax—INTEGER
- Access—Read-only
- Status—Mandatory
- Return value—Always returns 5

trapClientCount (1.3.6.1.3.94.2.2)

The current number of rows in the trap table.

- Syntax—INTEGER
- Access—Read-only
- Status—Mandatory
- Return value—Value (1-5) indicating number of configured trap clients

trapRegIpAddress (1.3.6.1.3.94.2.3.1.1)

The IP address of a client registered for traps.

- Syntax—IpAddress
- Access—Read-only
- Status—Mandatory
- Return value—IP addresses (as defined in the trap table) of where to send traps when they occur

trapRegPort (1.3.6.1.3.94.2.3.1.2)

The UDP port to which to send traps for this host. Normally this would be the standard trap port (162). This object is an index and must be specified to create a row in this table.

- Syntax—INTEGER (1..2147483647)
- Access—Read-only
- Status—Mandatory
- Return value—Configured port number to which to send traps when they occur. The port number can be configured in the switch SNMP setup parameters. Default is 162.

trapRegFilter (1.3.6.1.3.94.2.3.1.3)

This value defines the trap severity filter for this trap host. The connUnit sends traps to this host that have a severity level less than or equal to this value. The default value of this object is “warning”.

- Syntax—FcEventSeverity
- Access—Read-write
- Status—Mandatory
- Return value—Value indicating the trap severity level. Refer to [Table 28](#) for trap severity levels.

trapRegRowState (1.3.6.1.3.94.2.3.1.4)

Specifies the state of the row.

- rowDestroy
 - READ—Can never happen
 - WRITE—Remove this row from the table.
- rowInactive
 - READ—Indicates that this row does exist, but that traps are not enabled to be sent to the target.
 - WRITE—If the row does not exist, and the agent allows writes to the trap table, a new row is created. The values of the optional columns are set to default values. Traps are not enabled to be sent to the target. If the row already existed, traps are disabled from being sent to the target.

- rowActive
 - READ—Indicates that this row exists, and that traps are enabled to be sent to the target.
 - WRITE—If the row does not exist, and the agent allows writes to the trap table, a new row is created. The values of the optional columns are set to default values. Traps are enabled to be sent to the target. If the row already exists, traps are enabled to be sent to the target.

A value of “rowActive” or “rowInactive” must be specified to create a row in the table.
- Syntax—INTEGER { rowDestroy(1) - Remove row from table; rowInactive(2) - Row exists, but traps disabled; rowActive(3) - Row exists and is enabled for sending traps }
- Access—Read-write
- Status—Mandatory
- Return value—Returns rowActive (3), if valid entry in trap table

Related traps

The following traps contain the trap information being sent from the agent to the manager.

connUnitStatusChange (1.3.6.1.3.94.0.1)

The overall status of the connectivity unit has changed. The recommended severity level (for filtering) is “alert”. Sent whenever a Switch.OperChange or Switch.StateChange event occurs.

Variables: { connUnitStatus, connUnitState }

connUnitDeletedTrap (1.3.6.1.3.94.0.3)

A connUnit has been deleted from this agent. The recommended severity level (for filtering) is “warning”. Sent whenever an Eport.OperChange event occurs and the connUnitTable is smaller than previously noted (A connUnit has gone away).

Variables: { connUnitId }

connUnitEventTrap (1.3.6.1.3.94.0.4)

An event has been generated by the connectivity unit. The recommended severity level (for filtering) is “info”. Sent when a change notification occurs that does not fit into any other specific category.

Variables:

{ connUnitEventId, connUnitEventType, connUnitEventObject, connUnitEventDescr }

Table 29 lists the fields in the connUnitEventDescr variable.

Table 29: connUnitEventDescr Variable Field Descriptions

connUnitEventDescr variable	Description
Tag #	Number that identifies the event
Event #	Event counter
Chassis	Switch on which the event occurred
Blade	I/O blade on which the event occurred
Port	Port on which the event occurred
TimeDate	Time stamp of the event
Module	Software module where the event was initiated
Addr	Address in the software module where the event was initiated
Descr	Description of the event

Figure 2 provides the standard format of the connUnitEventDescr variable. Chassis, Blade, and Port are always 0.

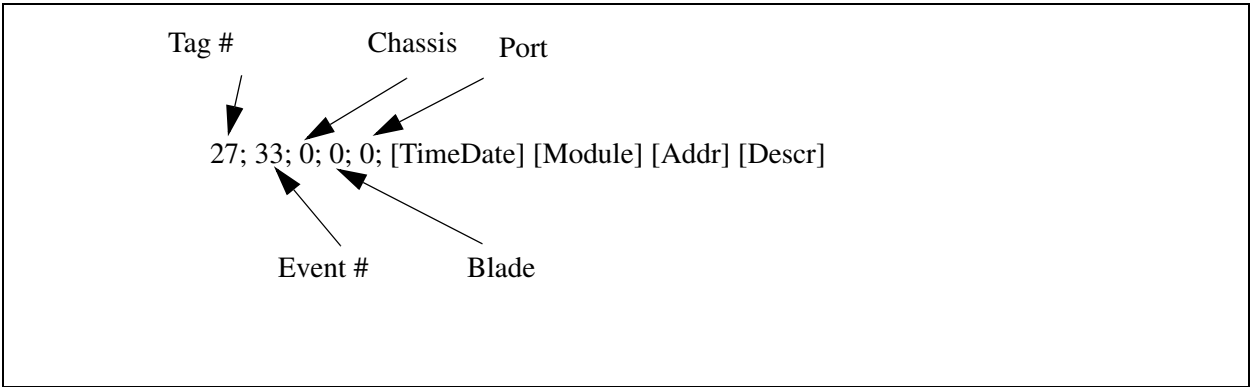


Figure 2: connUnitEventDescr variable format

Table 30 lists the possible trap strings returned for the connUnitEventDescr variable.

Table 30: connUnitEventDescr trap list

Trap type	Filter level	Notification
connUnitPortStatusChange	eventSeverity_info, eventSeverity_info, eventSeverity_critical, eventSeverity_info	UserPort.ConfigChange, UserPort.StateChange, Eport.ConvergeAlarm, UserPort.OperChange
connUnitDeletedTrap	eventSeverity_info, eventSeverity_info,	Fabric.OperChange, Eport.OperChange
connUnitStatusChange	eventSeverity_info, eventSeverity_info, eventSeverity_critical, eventSeverity_info	Switch.OperChange, Switch.StateChange, Switch.ResetAlarm, Chassis.OperChange

Table 30: connUnitEventDescr trap list (Continued)

Trap type	Filter level	Notification
connUnitSensorStatusChange	eventSeverity_critical, eventSeverity_critical, eventSeverity_critical, eventSeverity_critical, eventSeverity_critical, eventSeverity_critical, eventSeverity_critical	Chassis.PsBadAlarm, Chassis.PsOkAlarm, Chassis.FanBadAlarm, Chassis.FanOkAlarm, Blade.OverheatAlarm, Blade.OverwarmAlarm, Blade.TempOkAlarm
connUnitEventTrap	eventSeverity_info, eventSeverity_info, eventSeverity_info, eventSeverity_info, eventSeverity_info, eventSeverity_critical, eventSeverity_info, eventSeverity_warning, eventSeverity_warning, eventSeverity_critical, eventSeverity_info, eventSeverity_warning	Snmp.ConfigChange, Switch.ConfigChange, System.ConfigChange, Topology.OperChange, Zoning.Change, Zoning.MergeAlarm, NameServer.OperChange, Switch.AccessAlarm, Switch.AdminAlarm, Switch.GenericAlarm, Blade.OperChange, Switch.GenericEvent

connUnitSensorStatusChange (1.3.6.1.3.94.0.5)

The overall status of the connectivity unit has changed. The recommended severity level (for filtering) is “alert”. Sent whenever any of the following notifications occur:

- Chassis.PsBadAlarm
- Chassis.PsOkAlarm
- Chassis.FanBadAlarm
- Chassis.FanOkAlarm
- Blade.OverheatAlarm
- Blade.OverwarmAlarm

Variables: { connUnitSensorStatus }

connUnitPortStatusChange (1.3.6.1.3.94.0.6)

The overall status of the connectivity unit has changed. The recommended severity level (for filtering) is “alert”. Sent whenever a UserPort.StateChange or UserPort.OperChange event occurs.

Enterprise: fcmgmt

Variables: { connUnitPortStatus, connUnitPortState }

coldStart

A coldStart trap signifies that the SNMPv2 entity, acting in an agent role, is re-initializing itself and that its configuration may have been altered.

authenticationFailure

An authenticationFailure trap signifies that the SNMPv2 entity, acting in an agent role, has received a protocol message that is not properly authenticated. While all implementations of the SNMPv2 should be capable of generating this trap, the snmpEnableAuthenTraps object indicates whether this trap is generated.

Fabric Element MIB Objects

5

This chapter covers the implementation details for the Fabric Element Management Information Bases (FE-MIB) on the HP StorageWorks 2/8q FC Switch.

The following topics are covered:

- [Fibre Channel FE MIB definitions](#), page 142
- [Configuration group](#), page 143
- [Status group](#), page 149
- [Error group](#), page 155
- [Accounting groups](#), page 158
- [Capability group](#), page 164

Fibre Channel FE MIB definitions

The textual substitutions in [Table 31](#) are specific to the FE-MIB and can be used in place of primitive data types.

Table 31: FA-MIB textual substitutions

Description	Syntax
MilliSeconds	Unsigned32
MicroSeconds	Unsigned32
FcNameId	OCTET STRING (SIZE (8))
FcAddressId	OCTET STRING (SIZE (3))
FcRxDataFieldSize	Integer32 (128..2112)
FcBbCredit	Integer32 (0..32767)
FcphVersion	Integer32 (0..255)
FcStackedConnMode	INTEGER { none(1), transparent(2), lockedDown(3) }
FcCosCap	BITS { classF(0), class1(1), class2(2), class3(3), class4(4), class5(5), class6(6) }
FcFeModuleCapacity	Unsigned32
FcFeFxPortCapacity	Unsigned32
FcFeModuleIndex	Unsigned32
FcFeFxPortIndex	Unsigned32
FcFeNxPortIndex	Integer32 (1..126)
FcBbCreditModel	INTEGER { regular(1), alternate (2) }

Configuration group

The Configuration group represents a set of parameters associated with the Fabric Element or an FxPort to support its NxPorts. This group consists of scalar objects and the Module and FxPort Configuration tables.

Scalar objects

The objects described in this section are not in a table format. An example of how to access one of these objects is: “snmpget localhost public fcFeFabricName.0”

fcFeFabricName (1.3.6.1.2.1.75.1.1.1)

The Name_Identifier of the Fabric to which this Fabric Element belongs.

- Syntax—FcNameId
- Access—Read-write
- Status—Current
- Return value—World Wide Name of the switch (for example, 10 00 00 C0 DD 00 71 C9). Writes are not supported.

fcFeElementName (1.3.6.1.2.1.75.1.1.2)

The Name_Identifier of the Fabric Element.

- Syntax—FcNameId
- Access—Read-write
- Status—Current
- Return value—World Wide Name of the switch (for example, 10 00 00 C0 DD 00 71 C9). Writes are not supported.

fcFeModuleCapacity (1.3.6.1.2.1.75.1.1.3)

The maximum number of modules in the Fabric Element, regardless of their current state.

- Syntax—FcFeModuleCapacity
- Access—Read-only
- Status—Current
- Return value—Total number of switches in the fabric if ProxyEnable setting is Enabled on the out-of-band switch. If ProxyEnable setting is Disabled on the out-of-band switch, return value = 1.

Module table

The objects described in this section are in table format indexed by switch. An example of how to access one of these objects is: "snmpget localhost public fcFeModuleDescr.1". This table contains one entry for each module.

fcFeModuleDescr (1.3.6.1.2.1.75.1.1.4.1.2)

A textual description of the module. This value must include the full name and version identification of the module.

- Syntax—SnmpAdminString
- Access—Read-only
- Status—Current
- Return value—Configurable description of the module table entry. The default is HP StorageWorks 2/8q FC Switch.

fcFeModuleObjectID (1.3.6.1.2.1.75.1.1.4.1.3)

The vendor's authoritative identification of the module. This value can be allocated within the SMI enterprises subtree (1.3.6.1.4.1), and provides a means for determining what kind of module is being managed.

For example, this object could take the value 1.3.6.1.4.1.99649.3.9 if vendor "Neufe Inc." was assigned the subtree 1.3.6.1.4.1.99649, and had assigned the identifier 1.3.6.1.4.1.99649.3.9 to its FeFiFo-16 PlugInCard.

- Syntax—OBJECT IDENTIFIER
- Access—Read-only
- Status—Current
- Return value—Module identification number is HP StorageWorks 2/8q FC Switch = 1.3.6.1.4.1.1663.1.1.1.31.

fcFeModuleOperStatus (1.3.6.1.2.1.75.1.1.4.1.4)

Switch definitions map 1-to-1 with the MIB definitions. This object indicates the operational status of the module.

- online (1)—module is functioning properly
- offline (2)—module is not available
- testing (3)—module is under testing
- faulty (4)—module is defective in some way
 - Syntax—INTEGER { online(1) - functional; offline(2) - not available; testing(3) - under testing; faulty(4) - defective }
 - Access—Read-only
 - Status—Current
 - Return value—Operational status of that module

Table 32: Module operational status return values

Mode	Return value
Online	online(1)
Offline	offline(2)
Diagnostics	testing(3)
Other	faulty(4)

fcFeModuleLastChange (1.3.6.1.2.1.75.1.1.4.1.5)

This object contains the value of sysUpTime when the module entered its current operational status. A value of zero indicates that the operational status of the module has not changed since the agent last restarted.

- Syntax—TimeStamp
- Access—Read-only
- Status—Current
- Return value—Unsupported; always returns error status "NoSuchName"

fcFeModuleFxpPortCapacity (1.3.6.1.2.1.75.1.1.4.1.6)

The number of FxpPort that can be contained within the module. Within each module, the ports are uniquely numbered in the range from 1 to fcFeModuleFxpPortCapacity inclusive. However, the numbers are not required to be contiguous.

- Syntax—FcFeFxpPortCapacity
- Access—Read-only
- Status—Current
- Return value—Total number of ports capability on the switch. HP StorageWorks 2/8q FC Switch = 8.

fcFeModuleName (1.3.6.1.2.1.75.1.1.4.1.7)

The Name_Identifier of the switch.

- Syntax—FcNameId
- Access—Read-only
- Status—Current
- Return value—World Wide Name of the switch (for example, 10 00 00 C0 DD 00 71 C9). Writes are not supported.

FxPort configuration table

The objects described in this section are in table format indexed by switch index and port index. An example of how to access one of these objects is “nmpget localhost public fcFxPortName.1.1”. This table contains one entry for each FxPort and Configuration parameters of the ports.

fcFxPortName (1.3.6.1.2.1.75.1.1.5.1.2)

The World Wide Name of this FxPort. Each FxPort has a unique Port World Wide Name within the Fabric.

- Syntax—FcNameId
- Access—Read-only
- Status—Current
- Return value—Returns the Port World Wide Name for each port on switch. For example, the return value for port #2 would be 20 02 00 C0 DD 00 71 C9, and the return value for port #14 would be 20 0E 00 C0 DD 00 71 C9.

fcFxPortFcphVersionHigh (1.3.6.1.2.1.75.1.1.5.1.3)

The highest or most recent version of FC-PH that the FxPort is configured to support.

- Syntax—FcphVersion
- Access—Read-only
- Status—Current
- Return value—Always returns 32 (0x20)

fcFxPortFcphVersionLow (1.3.6.1.2.1.75.1.1.5.1.4)

The lowest or earliest version of FC-PH that the FxPort is configured to support.

- Syntax—FcphVersion
- Access—Read-only
- Status—Current
- Return value—Always returns 9

fcFxPortBbCredit (1.3.6.1.2.1.75.1.1.5.1.5)

The total number of receive buffers available for holding Class 1 connect-request, Class 2, or Class3 frames from the attached NxPort. It is for buffer-to-buffer flow control in the direction from the attached NxPort (if applicable) to FxPort.

- Syntax—FcBbCredit
- Access—Read-only
- Status—Current
- Return value—Default number of receive buffers for each port, unless extended credits are used. HP StorageWorks 2/8q FC Switch = 12.

fcFxpPortRxBufSize (1.3.6.1.2.1.75.1.1.5.1.6)

The largest Data_Field Size (in octets) for an FT_1 frame that can be received by the FxPort.

- Syntax—FcRxDataFieldSize
- Access—Read-only
- Status—Current
- Return value—Always returns 2112 (0x840)

fcFxpPortRatov (1.3.6.1.2.1.75.1.1.5.1.7)

The Resource_Allocation_Timeout Value configured for the FxPort. This is used as the timeout value for determining when to reuse an NxPort resource such as a Recovery_Qualifier. It represents E_D_TOV plus twice the maximum time that a frame can be delayed within the fabric and still be delivered. Refer to “[fcFxpPortEdtov \(1.3.6.1.2.1.75.1.1.5.1.8\)](#)” on page 147 for more information.

- Syntax—MilliSeconds
- Access—Read-only
- Status—Current
- Return value—Default is 10000 (0x2710)

fcFxpPortEdtov (1.3.6.1.2.1.75.1.1.5.1.8)

The E_D_TOV value configured for the FxPort. The Error_Detect_Timeout Value is used as the timeout value for detecting an error condition.

- Syntax—MilliSeconds
- Access—Read-only
- Status—Current
- Return value—Default is 2000 (0x7D0)

fcFxpPortCosSupported (1.3.6.1.2.1.75.1.1.5.1.9)

A value indicating the set of classes of service supported by the FxPort.

- Syntax—FcCosCap
- Access—Read-only
- Status—Current
- Return value—Always returns Class 3, 2, and F (0x0D)

fcFxpPortIntermixSupported (1.3.6.1.2.1.75.1.1.5.1.10)

A flag indicating whether or not the FxPort supports an Intermixed Dedicated Connection.

- Syntax—TruthValue
- Access—Read-only
- Status—Current
- Return value—Always returns False (2)

fcFxpPortStackedConnMode (1.3.6.1.2.1.75.1.1.5.1.11)

A value indicating the mode of Stacked Connect supported by the FxPort.

- Syntax—FcStackedConnMode
- Access—Read-only
- Status—Current
- Return value—Always returns None (1)

fcFxpPortClass2SeqDeliv (1.3.6.1.2.1.75.1.1.5.1.12)

A flag indicating whether or not Class 2 Sequential Delivery is supported by the FxPort.

- Syntax—TruthValue
- Access—Read-only
- Status—Current
- Return value—Always returns True (1)

fcFxpPortClass3SeqDeliv (1.3.6.1.2.1.75.1.1.5.1.13)

A flag indicating whether or not Class 3 Sequential Delivery is supported by the FxPort.

- Syntax—TruthValue
- Access—Read-only
- Status—Current
- Return value—Always returns True (1)

fcFxpPortHoldTime (1.3.6.1.2.1.75.1.1.5.1.14)

The maximum time, in microseconds, that the FxPort shall hold a frame before discarding the frame if it is unable to deliver the frame. The value 0 means that the FxPort does not support this parameter.

- Syntax—MicroSeconds
- Access—Read-only
- Status—Current
- Return value—Default ED_TOV parameter is 2000 (0x7D0)

Status group

These tables contain operational status and established service parameters for the Fabric Element and the attached NxPorts.

The status group consists of the following tables:

- FxPort status table
- FxPort physical table
- FxPort fabric login table

FxPort status table

This table contains one entry for each FxPort, its operational status, and parameters. The objects described in this section are in table format indexed by switch index and port index. An example of how to access one of these objects is: "snmpget localhost public fcFxPortId.1.1".

fcFxPortId (1.3.6.1.2.1.75.1.2.1.1.1)

The address identifier by which this FxPort is identified within the fabric. The FxPort can assign its address identifier to its attached NxPort(s) during Fabric Login.

- Syntax—FcAddressId
- Access—Read-only
- Status—Current
- Return value—Address of each port based on Domain, Area, and ALPA (for example, 64 03 00)

fcFxPortBbCreditAvailable (1.3.6.1.2.1.75.1.2.1.1.2)

The number of buffers currently available for receiving frames from the attached port in the buffer-to-buffer flow control. The value must be less than or equal to fcFxPortBbCredit.

- Syntax—Gauge32
- Access—Read-only
- Status—Current
- Return value—Unsupported; always returns error status "NoSuchName"

fcFxPortOperMode (1.3.6.1.2.1.75.1.2.1.1.3)

The current operational mode of the FxPort.

- Syntax—INTEGER { unknown(1); fPort(2); flPort(3) }
- Access—Read-only
- Status—Current
- Return value—Refer to [Table 33](#) for fcFxPortOperMode return values.

Table 33: Port operational modes

Mode	Return value
Unknown	1
F_Port	2
FL_Port	3

fcFxPortAdminMode (1.3.6.1.2.1.75.1.2.1.1.4)

The desired operational mode of the FxPort.

- Syntax—INTEGER { fPort(2), flPort(3) }
- Access—Read-write
- Status—Current
- Return value—Unsupported; always returns error status 'NoSuchName'

FxPort physical level table

The objects described in this section are in table format indexed by switch index and port index. An example of how to access one of these objects is: "snmpget localhost public fcFxPortPhysAdminStatus.1.1".

This table contains one entry for each FxPort in the Fabric Element, and the physical level status and parameters of the FxPorts.

fcFxPortPhysAdminStatus (1.3.6.1.2.1.75.1.2.2.1.1)

The desired state of the FxPort. A management station can place the FxPort in a desired state by setting this object accordingly. The testing(3) state indicates that no operational frames can be passed. When a Fabric Element initializes, all FxPorts start with fcFxPortPhysAdminStatus in the offline(2) state. As the result of either explicit management action or per configuration information accessible by the Fabric Element, fcFxPortPhysAdminStatus is changed to either the online(1) or testing(3) states, or remains in the offline state.

- Syntax—INTEGER { online(1) - place port online; offline(2) - take port offline; testing(3) - initiate test procedures }
- Access—Read-write
- Status—Current
- Return value—Refer to [Table 34](#) for fcFxPortPhysAdminStatus read values. Refer to [Table 35](#) for fcFxPortPhysAdminStatus write values.

Table 34: fcFxPortPhysAdminStatus read return values

Port	Value
Online	online (1)
Offline	offline (2)
Diagnostics	testing (3)
Down	offline (2)

Table 35: fcFxPortPhysAdminStatus write values

Port	Value
Online	online (1)
Offline	offline (2)
Diagnostics	testing (3)

fcFxPortPhysOperStatus (1.3.6.1.2.1.75.1.2.2.1.2)

The current operational status of the FxPort. The testing(3) indicates that no operational frames can be passed. If fcFxPortPhysAdminStatus is offline(2), fcFxPortPhysOperStatus must be offline(2). If fcFxPortPhysAdminStatus is changed to online(1), fcFxPortPhysOperStatus must change to online(1). If the FxPort is ready to accept Fabric Login request from the attached NxPort, it proceeds and remains in the link- failure(4) state if, and only if, there is a fault that prevents it from going to the online(1) state.

- Syntax—INTEGER { online(1) - Login can proceed; offline(2) - Login cannot proceed; testing(3) - port is under test; linkFailure(4) - failure after online/testing }
- Access—Read-only
- Status—Current
- Return value—Refer to [Table 36](#) for fcFxPortPhysOperStatus return values.

Table 36: fcFxPortPhysOperStatus return values

Status	Return value
Online	online (1)
Offline	offline (2)
Diagnostic	testing (3)
Down	linkfailure (4)

fcFxPortPhysLastChange (1.3.6.1.2.1.75.1.2.2.1.3)

The value of sysUpTime at the time the FxPort entered its current operational status. A value of zero indicates that the FxPort's operational status has not changed since the agent last restarted.

- Syntax—TimeStamp
- Access—Read-only
- Status—Current
- Return value—Unsupported; always returns error status "NoSuchName"

fcFxPortPhysRttov (1.3.6.1.2.1.75.1.2.2.1.4)

The Receiver_Transmitter_Timeout value of the FxPort. This is used by the receiver logic to detect a loss of synchronization.

- Syntax—Milliseconds
- Access—Read-write
- Status—Current
- Return value—Default RT_TOV parameter is: 100 (0x64). This is a global setting for the switch. If writing value to a port, all ports reflect this new value.

FxPort fabric login table

The objects described in this section are in table format indexed by switch index and port index. An example of how to access one of these objects is: "snmpget localhost public fcFxPortFcphVersionAgreed.1.1".

This table contains one entry for each FxPort in the fabric element, and the service parameters that have been established from the most recent Fabric Login, implicit or explicit.

fcFxPortFcphVersionAgreed (1.3.6.1.2.1.75.1.2.3.1.2)

The version of FC-PH that the FxPort has agreed to support from the Fabric Login.

- Syntax—FcphVersion
- Access—Read-only
- Status—Current
- Return value—Unsupported

fcFxPortNxPortBbCredit (1.3.6.1.2.1.75.1.2.3.1.3)

The total number of buffers available for holding class 1 connect-request, class 2, or class 3 frames to be transmitted to the attached NxPort. It is for buffer-to-buffer flow control in the direction from FxPort to NxPort. The buffer-to-buffer flow control mechanism is indicated in the respective fcFxPortBbCreditModel.

- Syntax—FcBbCredit
- Access—Read-only
- Status—Current
- Return value—Unsupported

fcFxPortNxPortRxDataFieldSize (1.3.6.1.2.1.75.1.2.3.1.4)

The Receive Data Field Size of the attached NxPort. This object specifies the largest Data Field Size for an FT_1 frame that can be received by the NxPort.

- Syntax—FcRxDataFieldSize
- Access—Read-only
- Status—Current
- Return value—Unsupported

fcFxPortCosSuppAgreed (1.3.6.1.2.1.75.1.2.3.1.5)

A variable indicating that the attached NxPort has requested the FxPort for the support of classes of services and the FxPort has granted the request.

- Syntax—FcCosCap
- Access—Read-only
- Status—Current
- Return value—Bits that have the following bit-mapped definition:
 - Bit 7—Class-six
 - Bit 6—Class-five
 - Bit 5—Class-four
 - Bit 4—Class-three
 - Bit 3—Class-two
 - Bit 2—Class-one
 - Bit 1—Class f

For example: If Class 3, return value 0x10

fcFxPortIntermixSuppAgreed (1.3.6.1.2.1.75.1.2.3.1.6)

A variable indicating that the attached NxPort has requested the FxPort for the support of Intermix and the FxPort has granted the request. This flag is only valid if Class 1 service is supported.

- Syntax—TruthValue
- Access—Read-only
- Status—Current
- Return value—Always returns false (2)

fcFxPortStackedConnModeAgreed (1.3.6.1.2.1.75.1.2.3.1.7)

A variable indicating whether the FxPort has agreed to support stacked connect from the Fabric Login. This is only meaningful if the ports are using Class 1 service.

- Syntax—FcStackedConnMode
- Access—Read-only
- Status—Current
- Return value—Always returns none (1)

fcFxpPortClass2SeqDelivAgreed (1.3.6.1.2.1.75.1.2.3.1.8)

A variable indicating whether the FxpPort has agreed to support Class 2 sequential delivery from the Fabric Login. This is only meaningful if the ports are using Class 2 service.

- Syntax—TruthValue
- Access—Read-only
- Status—Current
- Return value—Always returns true (1)

fcFxpPortClass3SeqDelivAgreed (1.3.6.1.2.1.75.1.2.3.1.9)

A flag indicating whether the FxpPort has agreed to support Class 3 sequential delivery from the Fabric Login. This is only meaningful if the ports are using Class 3 service.

- Syntax—TruthValue
- Access—Read-only
- Status—Current
- Return value—Always returns true (1)

fcFxpPortNxPortName (1.3.6.1.2.1.75.1.2.3.1.10)

The port name of the attached NxPort.

- Syntax—FcNameId
- Access—Read-only
- Status—Current
- Return value—Returns the Switch Port's Port World Wide Name for the attached device

fcFxpPortConnectedNxPort (1.3.6.1.2.1.75.1.2.3.1.11)

The address identifier of the destination NxPort with which this FxpPort is currently engaged in a either a Class 1 or loop connection. If this FxpPort is not engaged in a connection, the value of this object is "000000"H.

- Syntax—FcAddressId
- Access—Read-only
- Status—Current
- Return value—Unsupported

fcFxpPortBbCreditModel (1.3.6.1.2.1.75.1.2.3.1.12)

This object identifies the BB_Credit model used by the FxpPort.

- Syntax—FcBbCreditModel
- Access—Read-write
- Status—Current
- Return value—Returns alternate (2). Writes not supported.

Error group

This group consists of a single table (FxPort Error Table) that contains information about the various types of errors detected. The management station can use the information in this group to determine the quality of the link between the FxPort and its attached NxPort.

FxPort Error Table

The FxPort Error table contains counters that record numbers of errors detected since the management agent re-initialized for each FxPort in the Fabric Element. The objects described in this section are in table format indexed by switch index and port index. The first 6 columnar objects after the port index correspond to the counters in the Link Error Status Block. An example of how to access one of these objects is:

"snmpget localhost public fcFxPortLinkFailures.1.1".

fcFxPortLinkFailures (1.3.6.1.2.1.75.1.3.1.1.1)

The number of link failures detected by this FxPort.

- Syntax—Counter32
- Access—Read-only
- Status—Current
- Return value—Total number of LinkFailures encountered for a port

fcFxPortSyncLosses (1.3.6.1.2.1.75.1.3.1.1.2)

The number of loss of synchronizations detected by the FxPort.

- Syntax—Counter32
- Access—Read-only
- Status—Current
- Return value—Total number of LossOfSyncs encountered for a port

fcFxPortSigLosses (1.3.6.1.2.1.75.1.3.1.1.3)

The number of loss of signals detected by the FxPort.

- Syntax—Counter32
- Access—Read-only
- Status—Current
- Return value—Unsupported; always returns error status "NoSuchName".

fcFxPortPrimSeqProtoErrors (1.3.6.1.2.1.75.1.3.1.1.4)

The number of primitive sequence protocol errors detected by the FxPort.

- Syntax—Counter32
- Access—Read-only
- Status—Current
- Return value—Total number of PrimSeqErrors encountered for a port

fcFxpPortInvalidTxWords (1.3.6.1.2.1.75.1.3.1.1.5)

The number of invalid transmission words detected by the FxPort.

- Syntax—Counter32
- Access—Read-only
- Status—Current
- Return value—Total number of DecodeErrors encountered for a port

fcFxpPortInvalidCrcs (1.3.6.1.2.1.75.1.3.1.1.6)

The number of invalid CRCs detected by this FxPort.

- Syntax—Counter32
- Access—Read-only
- Status—Current
- Return value—Total number of InvalidCRCs encountered for a port

fcFxpPortDelimiterErrors (1.3.6.1.2.1.75.1.3.1.1.7)

The number of Delimiter Errors detected by this FxPort.

- Syntax—Counter32
- Access—Read-only
- Status—Current
- Return value—Unsupported; always returns error status "NoSuchName".

fcFxpPortAddressIdErrors (1.3.6.1.2.1.75.1.3.1.1.8)

The number of address identifier errors detected by this FxPort.

- Syntax—Counter32
- Access—Read-only
- Status—Current
- Return value—Total number of InvDestAddrs encountered for a port

fcFxpPortLinkResetIns (1.3.6.1.2.1.75.1.3.1.1.9)

The number of Link Reset Protocols received by this FxPort from the attached NxPort.

- Syntax—Counter32
- Access—Read-only
- Status—Current
- Return value—Total number of RxLinkResets received by a port

fcFxPortLinkResetOuts (1.3.6.1.2.1.75.1.3.1.1.10)

The number of Link Reset Protocols issued by this FxPort to the attached NxPort.

- Syntax—Counter32
- Access—Read-only
- Status—Current
- Return value—Total number of TxLinkResets sent by a port

fcFxPortOlsIns (1.3.6.1.2.1.75.1.3.1.1.11)

The number of Offline Sequences received by this FxPort.

- Syntax—Counter32
- Access—Read-only
- Status—Current
- Return value—Total number of RxOfflineSeqs received by a port

fcFxPortOlsOuts (1.3.6.1.2.1.75.1.3.1.1.12)

The number of Offline Sequences issued by this FxPort.

- Syntax—Counter32
- Access—Read-only
- Status—Current
- Return value—Total number of TxOfflineSeqs sent by a port

Accounting groups

The accounting group consists of the following tables that contain accounting information for FxPorts:

- Class 1 accounting table
- Class 2 accounting table
- Class 3 accounting table

Class 1 accounting table

The objects described in this section are in table format indexed by switch index and port index. An example of how to access one of these objects is: "snmpget localhost public fcFxPortC1InFrames.1.1". This table contains one entry for each FxPort in the Fabric Element and Counter32s for certain types of events that have occurred in the FxPorts since the management agent was re-initialized.

fcFxPortC1InFrames (1.3.6.1.2.1.75.1.4.1.1.1)

The number of Class 1 frames (other than Class 1 connect-request) received by this FxPort from its attached NxPort.

- Syntax—Counter32
- Access—Read-only
- Status—Current
- Return value—Unsupported; always returns error status "NoSuchName"

fcFxPortC1OutFrames (1.3.6.1.2.1.75.1.4.1.1.2)

The number of Class 1 frames (other than Class 1 connect-request) delivered through this FxPort to its attached NxPort.

- Syntax—Counter32
- Access—Read-only
- Status—Current
- Return value—Unsupported; always returns error status "NoSuchName"

fcFxPortC1InOctets (1.3.6.1.2.1.75.1.4.1.1.3)

The number of Class 1 frame octets, including the frame delimiters, received by this FxPort from its attached NxPort.

- Syntax—Counter32
- Access—Read-only
- Status—Current
- Return value—Unsupported; always returns error status "NoSuchName"

fcFxPortC1OutOctets (1.3.6.1.2.1.75.1.4.1.1.4)

The number of Class 1 frame octets, including the frame delimiters, delivered through this FxPort to its attached NxPort.

- Syntax—Counter32
- Access—Read-only
- Status—Current.
- Return value—Unsupported; always returns error status "NoSuchName"

fcFxPortC1Discards (1.3.6.1.2.1.75.1.4.1.1.5)

The number of Class 1 frames discarded by this FxPort.

- Syntax—Counter32
- Access—Read-only
- Status—Current
- Return value—Unsupported; always returns error status "NoSuchName"

fcFxPortC1FbsyFrames (1.3.6.1.2.1.75.1.4.1.1.6)

The number of F_BSY frames generated by this FxPort against Class 1 connect-request.

- Syntax—Counter32
- Access—Read-only
- Status—Current
- Return value—Unsupported; always returns error status "NoSuchName"

fcFxPortC1FrjtFrames (1.3.6.1.2.1.75.1.4.1.1.7)

The number of F_RJT frames generated by this FxPort against Class 1 connect-request.

- Syntax—Counter32
- Access—Read-only
- Status—Current
- Return value—Unsupported; always returns error status "NoSuchName"

fcFxPortC1InConnections (1.3.6.1.2.1.75.1.4.1.1.8)

The number of Class 1 connections successfully established in which the attached NxPort is the source of the connect-request.

- Syntax—Counter32
- Access—Read-only
- Status—Current
- Return value—Unsupported; always returns error status "NoSuchName"

fcFxPortC1OutConnections (1.3.6.1.2.1.75.1.4.1.1.9)

The number of Class 1 connections successfully established in which the attached NxPort is the destination of the connect-request.

- Syntax—Counter32
- Access—Read-only
- Status—Current
- Return value—Unsupported; always returns error status "NoSuchName"

fcFxPortC1ConnTime (1.3.6.1.2.1.75.1.4.1.1.10)

The cumulative time that this FxPort has been engaged in Class 1 connection. The amount of time is counted from after a connect-request has been accepted until the connection is disengaged, either by an EOFdt or Link Reset.

- Syntax—Milliseconds
- Access—Read-only
- Status—Current
- Return value—Unsupported; always returns error status "NoSuchName"

Class 2 accounting table

The objects described in this section are in table format indexed by switch index and port index. An example of how to access one of these objects is: "snmpget localhost public fcFxPortC2InFrames.1.1". This table contains one entry for each FxPort in the Fabric Element and Counter32s for certain types of events that have occurred in the FxPorts since the management agent was re-initialized.

fcFxPortC2InFrames (1.3.6.1.2.1.75.1.4.2.1.1)

The number of Class 2 frames received by this FxPort from its attached NxPort.

- Syntax—Counter32
- Access—Read-only
- Status—Current
- Return value—Total number of Class2FramesIn received by a port

fcFxPortC2OutFrames (1.3.6.1.2.1.75.1.4.2.1.2)

The number of Class 2 frames delivered through this FxPort to its attached NxPort.

- Syntax—Counter32
- Access—Read-only
- Status—Current
- Return value—Total number of Class2FramesOut sent by a port

fcFxPortC2InOctets (1.3.6.1.2.1.75.1.4.2.1.3)

The number of Class 2 frame octets, including the frame delimiters, received by this FxPort from its attached NxPort.

- Syntax—Counter32
- Access—Read-only
- Status—Current
- Return value—Total number of Class2WordsIn received by a port

fcFxPortC2OutOctets (1.3.6.1.2.1.75.1.4.2.1.4)

The number of Class 2 frame octets, including the frame delimiters, delivered through this FxPort to its attached NxPort.

- Syntax—Counter32
- Access—Read-only
- Status—Current
- Return value—Total number of Class2WordsOut sent by a port

fcFxPortC2Discards (1.3.6.1.2.1.75.1.4.2.1.5)

The number of Class 2 frames discarded by this FxPort.

- Syntax—Counter32
- Access—Read-only
- Status—Current
- Return value—Total number of Class2Toss discarded by a port

fcFxPortC2FbsyFrames (1.3.6.1.2.1.75.1.4.2.1.6)

The number of F_BSY frames generated by this FxPort.

- Syntax—Counter32
- Access—Read-only
- Status—Current
- Return value—Total number of FBusy frames generated by this port for Class 2 and 3 frames

fcFxPortC2FrjtFrames (1.3.6.1.2.1.75.1.4.2.1.7)

The number of F_RJT frames generated by this FxPort against Class 2 frames.

- Syntax—Counter32
- Access—Read-only
- Status—Current
- Return value—Total number of FReject frames generated by this port for Class 2 and 3 frames

Class 3 accounting table

The objects described in this section are in table format indexed by switch index and port index. An example of how to access one of these objects is: "snmpget localhost public fcFxPortC3InFrames.1.1". This table contains one entry for each FxPort in the Fabric Element and Counter32s for certain types of events that have occurred in the FxPorts since the management agent has re-initialized.

fcFxPortC3InFrames (1.3.6.1.2.1.75.1.4.3.1.1)

The number of Class 3 frames received by this FxPort from its attached NxPort.

- Syntax—Counter32
- Access—Read-only
- Status—Current
- Return value—Total number of Class3FramesIn received by a port

fcFxPortC3OutFrames (1.3.6.1.2.1.75.1.4.3.1.2)

The number of Class 3 frames delivered through this FxPort to its attached NxPort.

- Syntax—Counter32
- Access—Read-only
- Status—Current
- Return value—Total number of Class3FramesOut sent by a port

fcFxPortC3InOctets (1.3.6.1.2.1.75.1.4.3.1.3)

The number of Class 3 frame octets, including the frame delimiters, received by this FxPort from its attached NxPort.

- Syntax—Counter32
- Access—Read-only
- Status—Current
- Return value—Total number of Class3WordsOut received by a port

fcFxPortC3OutOctets (1.3.6.1.2.1.75.1.4.3.1.4)

The number of Class 3 frame octets, including the frame delimiters, delivered through this FxPort to its attached NxPort.

- Syntax—Counter32
- Access—Read-only
- Status—Current
- Return value—Total number of Class3WordsOut sent by a port

fcFxpPortC3Discards (1.3.6.1.2.1.75.1.4.3.1.5)

The number of Class 3 frames discarded by this FxPort.

- Syntax—Counter32
- Access—Read-only
- Status—Current
- Return value—Total number of Class3Toss discarded by a port

Capability group

The Capability Group consists of a single table (FxPort Capability Table) that describes FxPort capabilities. A capability can be used as expressed in its respective object value in the Configuration group.

FxPort capability table

The objects described in this section are in table format indexed by switch index and port index. An example of how to access one of these objects is: "snmpget localhost public fcFxPortCapFcphVersionHigh.1.1".

fcFxPortCapFcphVersionHigh (1.3.6.1.2.1.75.1.5.1.1.1)

The highest or most recent version of FC-PH that the FxPort is capable of supporting.

- Syntax—FcphVersion
- Access—Read-only
- Status—Current
- Return value—Always returns 32 (0x20)

fcFxPortCapFcphVersionLow (1.3.6.1.2.1.75.1.5.1.1.2)

The lowest or earliest version of FC-PH that the FxPort is capable of supporting.

- Syntax—FcphVersion
- Access—Read-only
- Status—Current
- Return value—Always returns 9

fcFxPortCapBbCreditMax (1.3.6.1.2.1.75.1.5.1.1.3)

The maximum number of receive buffers available for holding Class 1 connect-request, Class 2, or Class 3 frames from the attached NxPort.

- Syntax—FcBbCredit
- Access—Read-only
- Status—Current
- Return value—Default is 255 (0xFF)

fcFxPortCapBbCreditMin (1.3.6.1.2.1.75.1.5.1.1.4)

The minimum number of receive buffers available for holding Class 1 connect-request, Class 2, or Class 3 frames from the attached NxPort.

- Syntax—FcBbCredit
- Access—Read-only
- Status—Current
- Return value—Default is 0 (0x00)

fcFxpPortCapRxDataFieldSizeMax (1.3.6.1.2.1.75.1.5.1.1.5)

The maximum size in bytes of the Data Field in a frame that the FxPort is capable of receiving from its attached NxPort.

- Syntax—FcRxDataFieldSize
- Access—Read-only
- Status—Current
- Return value—2112 (0x840)

fcFxpPortCapRxDataFieldSizeMin (1.3.6.1.2.1.75.1.5.1.1.6)

The minimum size in bytes of the Data Field in a frame that the FxPort is capable of receiving from its attached NxPort.

- Syntax—FcRxDataFieldSize
- Access—Read-only
- Status—Current
- Return value—128 (0x80)

fcFxpPortCapCos (1.3.6.1.2.1.75.1.5.1.1.7)

A value indicating the set of classes of service that the FxPort is capable of supporting.

- Syntax—FcCosCap
- Access—Read-only
- Status—Current
- Return value—Always returns Class F, 2, and 3 (0x0d)

fcFxpPortCapIntermix (1.3.6.1.2.1.75.1.5.1.1.8)

A flag indicating whether or not the FxPort is capable of supporting the intermixing of Class 2 and Class 3 frames during a Class 1 connection. This flag is only valid if the port is capable of supporting Class 1 service.

- Syntax—TruthValue
- Access—Read-only
- Status—Current
- Return value—Always returns False (2)

fcFxpPortCapStackedConnMode (1.3.6.1.2.1.75.1.5.1.1.9)

A value indicating the mode of Stacked Connect request that the FxPort is capable of supporting.

- Syntax—FcStackedConnMode
- Access—Read-only
- Status—Current
- Return value—Always returns None (1)

fcFxpPortCapClass2SeqDeliv (1.3.6.1.2.1.75.1.5.1.1.10)

A flag indicating whether or not the FxPort is capable of supporting Class 2 Sequential Delivery.

- Syntax—TruthValue
- Access—Read-only
- Status—Current
- Return value—Always returns true (1)

fcFxpPortCapClass3SeqDeliv (1.3.6.1.2.1.75.1.5.1.1.11)

A flag indicating whether or not the FxPort is capable of supporting Class 3 Sequential Delivery.

- Syntax—TruthValue
- Access—Read-only
- Status—Current
- Return value—Always returns true (1)

fcFxpPortCapHoldTimeMaxv (1.3.6.1.2.1.75.1.5.1.1.12)

The maximum holding time that the FxPort is capable of supporting, in microseconds.

- Syntax—MicroSeconds
- Access—Read-only
- Status—Current
- Return value—20000 (0x4E20)

fcFxpPortCapHoldTimeMin (1.3.6.1.2.1.75.1.5.1.1.13)

The minimum holding time that the FxPort is capable of supporting, in microseconds.

- Syntax—MicroSeconds
- Access—Read-only
- Status—Current
- Return value—10 (0x0A)

MIB Objects

6

This chapter covers the implementation details for the Management Information Bases (MIB) on the HP StorageWorks 2/8q Fibre Channel Switch.

MIB definitions

This MIB replaces the fcFxpPortPhysTable module defined in FIBRE-CHANNEL-FE-MIB, and defines volatile control objects for ports in a switch. If the switch gets reset, these values revert back to the default values in the configuration file.

fcQxPortPhysAdminStatus (1.3.6.1.4.1.1663.1.3.10.1.1.3)

The desired state of the FxPort. A management station can place the FxPort in a desired state by setting this object accordingly. The testing(3) state indicates that no operational frames can be passed. When a Fabric Element initializes, all FxPorts start with fcQxPortPhysAdminStatus in the offline(2) state. As the result of either explicit management action or per configuration information accessible by the Fabric Element, fcQxPortPhysAdminStatus is changed to either the online(1) or testing(3) states, or remains in the offline state.

- Syntax—INTEGER { online(1) - place port online; offline(2) - take port offline; testing(3) - initiate test procedures }
- Access—Read-write
- Status—Current
- Return value—Refer to [Table 37](#) for fcQxPortPhysAdminStatus read values. Refer to [Table 38](#) for fcQxPortPhysAdminStatus write values.

Table 37: fcQxPortPhysAdminStatus read return values

Port	Value
Online	online (1)
Offline	offline (2)
Diagnostics	testing (3)
Down	offline (2)

Table 38: fcQxPortPhysAdminStatus write values

Port	Value
Online	online (1)
Offline	offline (2)
Diagnostics	testing (3)

fcQxPortPhysOperStatus (1.3.6.1.4.1.1663.1.3.10.1.1.4)

The current operational status of the FxPort. The testing(3) indicates that no operational frames can be passed. If fcQxPortPhysAdminStatus is offline(2), fcQxPortPhysOperStatus must be offline(2). If fcQxPortPhysAdminStatus is changed to online(1), fcQxPortPhysOperStatus changes to online(1). If the FxPort is ready to accept Fabric Login request from the attached NxPort, it proceeds and remains in the link- failure(4) state if, and only if, there is a fault that prevents it from going to the online(1) state.

- Syntax—INTEGER { online(1) - Login can proceed; offline(2) - Login cannot proceed; testing(3) - port is under test; linkFailure(4) - failure after online/testing }
- Access—Read-only
- Status—Current
- Return value—Refer to [Table 39](#) for fcQxPortPhysOperStatus return values.

Table 39: fcFxPortPhysOperStatus return values

Status	Return value
Online	online (1)
Offline	offline (2)
Diagnostic	testing (3)
Down	linkfailure (4)

Index

A

Accounting Groups [162](#)
Additional IP Objects [54](#)
Additional TCP Objects [63](#)
Address Translation Group [44](#)
Agent [25](#)
Alert [26](#), [138](#)
atIfIndex [44](#)
atNetAddress [44](#)
atPhysAddress [44](#)
atTable [44](#)
audience [16](#)
authorized reseller, HP [21](#)

C

Capability Group [168](#)
Class 1 Accounting Table [162](#)
Class 2 Accounting Table [164](#)
Class 3 Accounting Table [166](#)
Configuration Group [147](#)
configurationChangeTime [80](#)
Configuring switch [33](#)
Connectivity Table [82](#)
Connectivity Unit Group [80](#)
connUnitConfigurationChangeTime [86](#)
connUnitContact [88](#)
connUnitControl [87](#)
connUnitDomainId [85](#)
connUnitEventCurrId [90](#)
connUnitEventDescr [109](#)
connUnitEventFilter [89](#)
connUnitEventId [107](#)
connUnitEventIndex [106](#)
connUnitEventObject [109](#)
connUnitEventSeverity [108](#)
connUnitEventType [109](#)
connUnitEventUnitId [106](#)
connUnitGlobalId [82](#)
connUnitId [82](#)

connUnitInfo [87](#)
connUnitLinkAgentAddressTypeY [112](#)
connUnitLinkAgentAddressY [112](#)
connUnitLinkAgentPortY [113](#)
connUnitLinkConnIdY [113](#)
connUnitLinkCurrIndex [113](#)
connUnitLinkIndex [110](#)
connUnitLinkNodeIdX [111](#)
connUnitLinkNodeIdY [111](#)
connUnitLinkPortNumber [112](#)
connUnitLinkPortNumberX [111](#)
connUnitLinkPortWwnX [111](#)
connUnitLinkPortWwnY [112](#)
connUnitLinkUnitId [110](#)
connUnitLinkUnitTypeY [113](#)
connUnitLocation [88](#)
connUnitMaxEvents [90](#)
connUnitModuleId [87](#)
connUnitName [87](#)
connUnitNumEvents [89](#)
connUnitNumports [82](#)
connUnitNumRevs [86](#)
connUnitNumSensors [85](#)
connUnitNumZones [86](#)
connUnitPortControl [102](#)
connUnitPortFCClassCap [98](#)
connUnitPortFCClassOp [98](#)
connUnitPortFCId [101](#)
connUnitPortHWState [105](#)
connUnitPortIndex [97](#)
connUnitPortModuleType [100](#)
connUnitPortName [103](#)
connUnitPortNodeWwn [104](#)
connUnitPortPhysicalNumber [103](#)
connUnitPortProtocolCap [104](#)
connUnitPortProtocolOp [104](#)
connUnitPortRevision [101](#)
connUnitPortSn [101](#)
connUnitPortSpeed [101](#)
connUnitPortStatCountAddressErrors [130](#)

- connUnitPortStatCountBBCreditZero 121
- connUnitPortStatCountClass1FBSYFrames 123
- connUnitPortStatCountClass1FRJTFrames 123
- connUnitPortStatCountClass1PBSYFrames 123
- connUnitPortStatCountClass1PRJTFrames 123
- connUnitPortStatCountClass1RxFrames 122
- connUnitPortStatCountClass1TxFrames 122
- connUnitPortStatCountClass2FBSYFrames 124
- connUnitPortStatCountClass2FRJTFrames 125
- connUnitPortStatCountClass2PBSYFrames 124
- connUnitPortStatCountClass2PRJTFrames 125
- connUnitPortStatCountClass2RxFrames 124
- connUnitPortStatCountClass2TxFrames 124
- connUnitPortStatCountClass3Discards 125
- connUnitPortStatCountClass3RxFrames 125
- connUnitPortStatCountClass3TxFrames 125
- connUnitPortStatCountDelimiterErrors 130
- connUnitPortStatCountEncodingDisparityErrors 130
- connUnitPortStatCountError 119
- connUnitPortStatCountFBSYFrames 121
- connUnitPortStatCountFramesTooLong 129
- connUnitPortStatCountFramesTruncated 130
- connUnitPortStatCountFRJTFrames 122
- connUnitPortStatCountInputBuffersFull 121
- connUnitPortStatCountInvalidCRC 128
- connUnitPortStatCountInvalidOrderedSets 129
- connUnitPortStatCountInvalidTxWords 128
- connUnitPortStatCountLinkFailures 128
- connUnitPortStatCountLossOfSignal 129
- connUnitPortStatCountLossOfSynchronization 129
- connUnitPortStatCountNumberLinkResets 127
- connUnitPortStatCountNumberOfflineSequences 127
- connUnitPortStatCountPBSYFrames 121
- connUnitPortStatCountPrimitiveSequenceProtocolErrors 128
- connUnitPortStatCountPRJTFrames 122
- connUnitPortStatCountRxBroadcastObjects 126
- connUnitPortStatCountRxElements 120
- connUnitPortStatCountRxLinkResets 126
- connUnitPortStatCountRxMulticastObjects 126
- connUnitPortStatCountRxObjects 120
- connUnitPortStatCountRxOfflineSequences 127
- connUnitPortStatCountTxBroadcastObjects 126
- connUnitPortStatCountTxElements 120
- connUnitPortStatCountTxLinkResets 127
- connUnitPortStatCountTxMulticastObjects 126
- connUnitPortStatCountTxObjects 120
- connUnitPortStatCountTxOfflineSequences 127
- connUnitPortState1 99
- connUnitPortStatIndex 119
- connUnitPortStatObject 103
- connUnitPortStatUnitId 119
- connUnitPortStatus 99
- connUnitPortTransmitterType 99
- connUnitPortType 97
- connUnitPortUnitId 97
- connUnitPortWwn 100
- connUnitPrincipal 85
- connUnitProduct 84
- connUnitProxyMaster 85
- connUnitREventTime 107
- connUnitRevsDescription 92
- connUnitRevsIndex 91
- connUnitRevsRevId 91
- connUnitRevsUnitId 91
- connUnitSensorCharacteristic 96
- connUnitSensorIndex 93
- connUnitSensorInfo 94
- connUnitSensorMessage 95
- connUnitSensorName 93
- connUnitSensorStatus 94
- connUnitSensorType 96
- connUnitSensorUnitId 93
- connUnitSEventTime 107
- connUnitSn 84
- connUnitSnsClassOfSvc 132
- connUnitSnsFabricPortName 134
- connUnitSnsFC4Type 133
- connUnitSnsHardAddress 134
- connUnitSnsId 131
- connUnitSnsNodeIPAddress 132
- connUnitSnsNodeName 132
- connUnitSnsPortIdentifier 132
- connUnitSnsPortIndex 131
- connUnitSnsPortIPAddress 134
- connUnitSnsPortName 132
- connUnitSnsPortType 133
- connUnitSnsProcAssoc 133
- connUnitSnsSymbolicNodeName 134
- connUnitSnsSymbolicPortName 134
- connUnitState 83
- connUnitStatus 83
- connUnitStatusChangeTime 86
- connUnitTableChangeTime 81
- connUnitType 82
- connUnitUpTime 84
- connUnitUrl 84
- conventions
 - document 17
 - equipment symbols 18
 - text symbols 17
- Critical 26, 138
- D**
 - Debug 26, 138
 - document
 - conventions 17
 - prerequisites 16

related documentation 16

E

EGP Group 66
 EGP Neighbor Table 66
 egpAs 69
 egpInErrors 66
 egpInMsgs 66
 egpNeighAddr 67
 egpNeighAs 67
 egpNeighEventTrigger 69
 egpNeighInErrMsgs 68
 egpNeighInErrs 67
 egpNeighInMsgs 67
 egpNeighIntervalHello 68
 egpNeighIntervalPoll 68
 egpNeighMode 69
 egpNeighOutErrMsgs 68
 egpNeighOutErrs 67
 egpNeighOutMsgs 67
 egpNeighState 66
 egpNeighStateDowns 68
 egpNeighStateUps 68
 egpNeighTable 66
 egpOutErrors 66
 egpOutMsgs 66
 Emergency 26, 138
 equipment symbols 18
 Error 26, 138
 Error Group 159
 Event Table 106

F

fcFeElementName 147
 fcFeFabricName 147
 fcFeModuleCapacity 147
 fcFeModuleDescr 148
 fcFeModuleFxpPortCapacity 149
 fcFeModuleLastChange 149
 fcFeModuleName 149
 fcFeModuleObjectID 148
 fcFeModuleOperStatus 148
 fcFeModuleTable 148
 fcFxpPortAddressIdErrors 160
 fcFxpPortAdminMode 154
 fcFxpPortBbCredit 150
 fcFxpPortBbCreditAvailable 153
 fcFxpPortBbCreditModel 158
 fcFxpPortC1AccountingTable 162
 fcFxpPortC1ConnTime 164
 fcFxpPortC1Discards 163
 fcFxpPortC1FbsyFrames 163
 fcFxpPortC1FrtFrames 163

fcFxpPortC1InConnections 163
 fcFxpPortC1InFrames 162
 fcFxpPortC1InOctets 162
 fcFxpPortC1OutConnections 164
 fcFxpPortC1OutFrames 162
 fcFxpPortC1OutOctets 163
 fcFxpPortC2AccountingTable 164
 fcFxpPortC2Discards 165
 fcFxpPortC2FbsyFrames 165
 fcFxpPortC2FrtFrames 165
 fcFxpPortC2InFrames 164
 fcFxpPortC2InOctets 165
 fcFxpPortC2OutFrames 164
 fcFxpPortC2OutOctets 165
 fcFxpPortC3Discards 167
 fcFxpPortC3InFrames 166
 fcFxpPortC3InOctets 166
 fcFxpPortC3OutFrames 166
 fcFxpPortC3OutOctets 166
 fcFxpPortCapBbCreditMax 168
 fcFxpPortCapBbCreditMin 168
 fcFxpPortCapClass2SeqDeliv 170
 fcFxpPortCapClass3SeqDeliv 170
 fcFxpPortCapCos 169
 fcFxpPortCapFcphVersionHigh 168
 fcFxpPortCapFcphVersionLow 168
 fcFxpPortCapHoldTimeMax 170
 fcFxpPortCapHoldTimeMin 170
 fcFxpPortCapIntermix 169
 fcFxpPortCapRxDataFieldSizeMax 169
 fcFxpPortCapRxDataFieldSizeMin 169
 fcFxpPortCapStackedConnMode 169
 fcFxpPortCapTable 168
 fcFxpPortClass2SeqDeliv 152
 fcFxpPortClass2SeqDelivAgreed 158
 fcFxpPortClass3SeqDeliv 152
 fcFxpPortClass3SeqDelivAgreed 158
 fcFxpPortConnectedNxPort 158
 fcFxpPortCosSuppAgreed 157
 fcFxpPortCosSupported 151
 fcFxpPortDelimiterErrors 160
 fcFxpPortEdtov 151
 fcFxpPortFcphVersionAgreed 156
 fcFxpPortFcphVersionHigh 150
 fcFxpPortFcphVersionLow 150
 fcFxpPortHoldTime 152
 fcFxpPortID 153
 fcFxpPortIntermixSuppAgreed 157
 fcFxpPortIntermixSupported 151
 fcFxpPortInvalidCrcs 160
 fcFxpPortInvalidTxWords 160
 fcFxpPortLinkFailures 159
 fcFxpPortLinkResetIns 160
 fcFxpPortLinkResetOuts 161

- fcFxpPortName 150
- fcFxpPortNxPortBbCredit 156
- fcFxpPortNxPortName 158
- fcFxpPortNxPortRxDataFieldSize 157
- fcFxpPortOlIns 161
- fcFxpPortOlSOuts 161
- fcFxpPortOperMode 153
- fcFxpPortPhysAdminStatus 154
- FcFxpPortPhysEntry 154
- fcFxpPortPhysLastChange 156
- fcFxpPortPhysOperStatus 155, 173
- fcFxpPortPhysRttov 156
- fcFxpPortPhysTable 154
- fcFxpPortPrimSeqProtoErrors 159
- fcFxpPortRatov 151
- fcFxpPortRxBufSize 151
- fcFxpPortSigLosses 159
- fcFxpPortStackedConnMode 152
- fcFxpPortStackedConnModeAgreed 157
- fcFxpPortSyncLosses 159
- FxPort Capability Table 168
- FxPort Configuration Table 150
- FxPort Error Table 159
- FxPort Fabric Login Table 156
- FxPort Physical Level Table 154
- FxPort physical level table 154
- FxPort status table 153

G

- getting help 20
- Groups in MIB-II 36

H

- help, obtaining 20
- HP
 - authorized reseller 21
 - storage web site 21
 - technical support 20

I

- ICMP Group 55
- icmplnAddrMaskReps 57
- icmplnAddrMasks 57
- icmplnDestUnreachs 55
- icmplnEchoReps 56
- icmplnEchos 56
- icmplnErrors 55
- icmplnMsgs 55
- icmplnParmProbs 55
- icmplnRedirects 56
- icmplnSrcQuenchs 56
- icmplnTimeExcds 55
- icmplnTimestampReps 56
- icmplnTimestamps 56
- icmpOutAddrMaskReps 59
- icmpOutAddrMasks 59
- icmpOutDestUnreachs 57
- icmpOutEchoReps 58
- icmpOutEchos 58
- icmpOutErrors 57
- icmpOutMsgs 57
- icmpOutParmProbs 58
- icmpOutRedirects 58
- icmpOutSrcQuenchs 58
- icmpOutTimeExcds 57
- icmpOutTimestampReps 59
- icmpOutTimestamps 58
- ifAdminStatus 40
- ifDescr 39
- ifIndex 39
- ifInDiscards 41
- ifInErrors 41
- ifInNUcastPkts 41
- ifInOctets 41
- ifInUcastPkts 41
- ifInUnknownProtos 42
- ifLastChange 41
- ifMtu 40
- ifNumber 39
- ifOperStatus 40
- ifOutDiscards 42
- ifOutErrors 42
- ifOutNUcastPkts 42
- ifOutOctets 42
- ifOutQLen 43
- ifOutUcastPkts 42
- ifPhysAddress 40
- ifSpecific 43
- ifSpeed 40
- ifType 39
- Info 26, 138
- Interfaces Group 39
- Interfaces Table 39
- IP Address Table 49
- IP Group 45
- IP Routing Table 50
- ipAddrTable 49
- ipAdEntAddr 49
- ipAdEntBcastAddr 49
- ipAdEntIfIndex 49
- ipAdEntNetMask 49
- ipAdEntReasmMaxSize 49
- ipDefaultTTL 45
- ipForwarding 45
- ipForwDatagrams 46
- ipFragCreates 48
- ipFragFails 48

[ipFragOKs 48](#)
[ipInAddrErrors 46](#)
[ipInDelivers 46](#)
[ipInDiscards 46](#)
[ipInHdrErrors 45](#)
[ipInReceives 45](#)
[ipInUnknownProtos 46](#)
[ipNetToMediaIfIndex 53](#)
[ipNetToMediaNetAddress 53](#)
[ipNetToMediaPhysAddress 53](#)
[ipNetToMediaType 53](#)
[ipOutDiscards 47](#)
[ipOutNoRoutes 47](#)
[ipOutRequests 47](#)
[ipReasmFails 48](#)
[ipReasmOKs 48](#)
[ipReasmReqs 47](#)
[ipReasmTimeout 47](#)
[ipRouteAge 52](#)
[ipRouteDest 50](#)
[ipRouteIfIndex 50](#)
[ipRouteInfo 52](#)
[ipRouteMask 52](#)
[ipRouteMetric1 50](#)
[ipRouteMetric2 50](#)
[ipRouteMetric3 51](#)
[ipRouteMetric4 51](#)
[ipRouteMetric5 52](#)
[ipRouteNextHop 51](#)
[ipRouteProto 52](#)
[ipRouteTable 50](#)
[ipRouteType 51](#)
[ipRoutingDiscards 54](#)

L

[Link Table 110](#)

M

[Management information bases 35](#)
[Manager 25](#)
[Mark 138](#)
[MIB Definitions 146](#)
[MIB-II 35](#)
[Module Table 148](#)

N

[Notify 26, 138](#)

P

[Port Statistics Table 119](#)
[Port Table 97](#)
[prerequisites 16](#)

R

[rack stability, warning 19](#)
[related documentation 16](#)
[Revision Table 91](#)
[revisionNumber 79](#)

S

[scalar objects 147](#)
[Sensor Table 93](#)
[Simple Name Server Table 131](#)
[Simple Network Management Protocol 23](#)
[SNMP 23](#)
[SNMP Group 71](#)
[snmpEnableAuthenTraps 76](#)
[snmpInASNParseErrs 72](#)
[snmpInBadCommunityNames 71](#)
[snmpInBadCommunityUses 72](#)
[snmpInBadValues 72](#)
[snmpInBadVersions 71](#)
[snmpInGenErrs 73](#)
[snmpInGetNexts 74](#)
[snmpInGetRequests 73](#)
[snmpInGetResponses 74](#)
[snmpInNoSuchNames 72](#)
[snmpInPkts 71](#)
[snmpInReadOnlys 73](#)
[snmpInSetRequests 74](#)
[snmpInTooBig 72](#)
[snmpInTotalReqVars 73](#)
[snmpInTotalSetVars 73](#)
[snmpInTraps 74](#)
[snmpOutBadValues 75](#)
[snmpOutGenErrs 75](#)
[snmpOutGetNexts 75](#)
[snmpOutGetRequests 75](#)
[snmpOutGetResponses 76](#)
[snmpOutNoSuchNames 75](#)
[snmpOutPkts 71](#)
[snmpOutSetRequests 75](#)
[snmpOutTooBig 74](#)
[snmpOutTraps 76](#)
[SNMPv1 24](#)
[SNMPv2c 24](#)
[statusChangeTime 80](#)
[Switch Manager 33](#)
[symbols in text 17](#)
[symbols on equipment 18](#)
[sysContact 38](#)
[sysDescr 37](#)
[sysLocation 38](#)
[sysName 38](#)
[sysObjectID 37](#)
[sysServices 38](#)

System Group [37](#)
systemURL [80](#)
sysUpTime [37](#)

T

TCP Connection Table [62](#)
TCP Group [60](#)
tcpActiveOpens [61](#)
tcpAttemptFails [61](#)
tcpConnLocalAddress [63](#)
tcpConnLocalPort [63](#)
tcpConnRemAddress [63](#)
tcpConnRemPort [63](#)
tcpConnState [62](#)
tcpCurrEstab [61](#)
tcpEstabResets [61](#)
tcpInErrs [63](#)
tcpInSegs [62](#)
tcpMaxConn [60](#)
tcpOutRsts [63](#)
tcpOutSegs [62](#)
tcpPassiveOpens [61](#)
tcpRetransSegs [62](#)
tcpRtoAlgorithm [60](#)
tcpRtoMax [60](#)
tcpRtoMin [60](#)
technical support, HP [20](#)
text symbols [17](#)

Transmission Group [70](#)
trap severity levels [26](#), [138](#)
Trap Table [138](#)
trapRegFilter [139](#)
trapRegIpAddress [139](#)
trapRegPort [139](#)
trapRegRowState [139](#)

U

UDP Group [64](#)
UDP Listener Table [64](#)
udpInDatagrams [64](#)
udpInErrors [64](#)
udpLocalAddress [64](#)
udpLocalPort [65](#)
udpNoPorts [64](#)
udpOutDatagrams [64](#)
Unknown [26](#), [138](#)
uNumber [80](#)

W

Warning [26](#), [138](#)
warning
 rack stability [19](#)
 symbols on equipment [18](#)
web sites
 HP storage [21](#)